COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

03/2019 VOL.62 NO.03

The Seven Tools of Causal Inference with Reflections on Machine Learning

Owning Computing's Environmental Impact

Beyond Worst-Case Analysis

Building a Better Battery

Association for Computing Machinery

Trust Extension as a Mechanism for Secure Code Execution on Commodity Computers
Bryan Jeffrey Parno
Text Data Management and Analysis
A Practical Introduction to Information Retrieval and Text Mining
ChengXiang Zhai
Sean Massung
The Sparse Fourier Transform
Theory and Practice
Haitham Hassanieh
Frontiers of Multimedia Research
Shih-Fu Chang
Editor
Smarter than Their Machines
Oral Histories of Pioneers in Interactive Computing
John Cullinane
The VR Book
Human-Centered Design for Virtual Reality
Jason Jerald, Ph.D.
Verified Functional Programming in Agda
Aaron Stump
A Framework for Scientific Discovery through Video Games
Seth Cooper
An Architecture for Fast and General Data Processing on Large Clusters
Matei Zaharia
The Handbook of Multimodal-Multisensor Interfaces
Volume 1
Foundations, User Modeling, and Common Modality Combinations

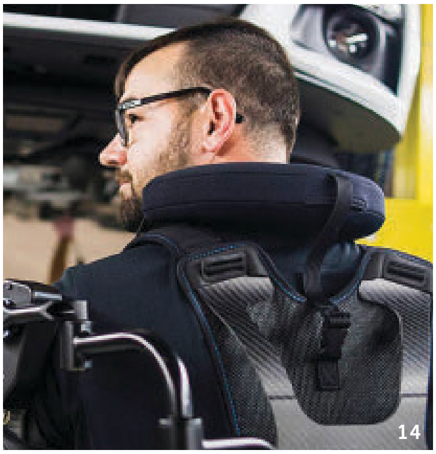
COMMUNICATIONS OF THE ACM

Departments

Last Byte

Association for Computing Machinery
Advancing Computing as a Science & Profession

News



Viewpoints

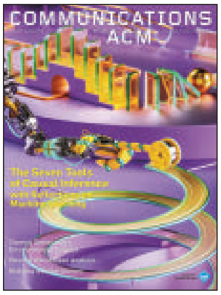
IMAGE COURTESY OF EKSO BIONICS AND FORD

03/2019
VOL. 62 NO. 03

Practice

Articles' development led by acmqueue
queue.acm.org

IMAGES BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK



About the Cover:
Turing Award recipient Judea Pearl argues (p. 54) that causal reasoning, an indispensible building block of human thought, must be incorporated into machine learning. He offers several innovative tools that may help ML realize causal inference. Cover illustration by Vault49.

Contributed Articles

Review Articles

Research Highlights

COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

Communications of the ACM is the leading monthly print and online magazine for the computing and information technology fields. *Communications* is recognized as the most trusted and knowledgeable source of industry information for today's computing professional. *Communications* brings its readership in-depth coverage of emerging areas of computer science, new trends in information technology, and practical applications. Industry leaders use *Communications* as a platform to present and debate various technology implications, public policies, engineering challenges, and market trends. The prestige and unmatched reputation that *Communications of the ACM* enjoys today is built upon a 50-year commitment to high-quality editorial content and a steadfast dedication to advancing the arts, sciences, and applications of information technology.

ACM, the world's largest educational and scientific computing society, delivers resources that advance computing as a science and profession. ACM provides the computing field's premier Digital Library and serves its members and the computing profession with leading-edge publications, conferences, and career resources.

ACM U.S. Technology Policy Office
Adam Eisgrau,
Director of Global Policy and Public Affairs
1701 Pennsylvania Ave NW, Suite 200,
Washington, DC 20006 USA
T (202) 580-6555; acmpo@acm.org

Computer Science Teachers Association
Jake Baskin
Executive Director

STAFF
DIRECTOR OF PUBLICATIONS
Scott E. Delman
cacm-publisher@cacm.acm.org

Executive Editor
Diane Crawford
Managing Editor
Thomas E. Lambert
Senior Editor
Andrew Rosenbloom
Senior Editor/News
Lawrence M. Fisher
Web Editor
David Roman
Editorial Assistant
Danbi Yu

Art Director
Andrij Borys
Associate Art Director
Margaret Gray
Assistant Art Director
Mia Angelica Balaquiot
Production Manager
Bernadette Shade
Intellectual Property Rights Coordinator
Barbara Ryan
Advertising Sales Account Manager
Ilia Rodriguez

Columnists
David Anderson; Michael Cusumano; Peter J. Denning; Mark Guzdial; Thomas Haigh; Leah Hoffmann; Mari Sako; Pamela Samuelson; Marshall Van Alstyne

CONTACT POINTS
Copyright permission
permissions@hq.acm.org
Calendar items
calendar@cacm.acm.org
Change of address
acmhelp@acm.org
Letters to the Editor
letters@cacm.acm.org

WEBSITE
http://cacm.acm.org

WEB BOARD
Chair
James Landay
Board Members
Marti Hearst; Jason I. Hong; Jeff Johnson; Wendy E. MacKay

AUTHOR GUIDELINES
http://cacm.acm.org/about-communications/author-center

ACM ADVERTISING DEPARTMENT
2 Penn Plaza, Suite 701, New York, NY 10121-0701
T (212) 626-0686
F (212) 869-0481

Advertising Sales Account Manager
Ilia Rodriguez
ilia.rodriguez@hq.acm.org

Media Kit acmmediasales@acm.org

Association for Computing Machinery (ACM)
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA
T (212) 869-7440; F (212) 869-0481

Subscriptions
An annual subscription cost is included in ACM member dues of $99 ($40 of which is allocated to a subscription to *Communications*); for students, cost is included in $42 dues ($20 of which is allocated to a *Communications* subscription). A nonmember annual subscription is $269.

ACM Media Advertising Policy
Communications of the ACM and other ACM Media publications accept advertising in both print and electronic formats. All advertising in ACM Media publications is at the discretion of ACM and is intended to provide financial support for the various activities and services for ACM members. Current advertising rates can be found by visiting http://www.acm-media.org or by contacting ACM Media Sales at (212) 626-0686.

Single Copies
Single copies of *Communications of the ACM* are available for purchase. Please contact acmhelp@acm.org.

COMMUNICATIONS OF THE ACM
(ISSN 0001-0782) is published monthly by ACM Media, 2 Penn Plaza, Suite 701, New York, NY 10121-0701. Periodicals postage paid at New York, NY 10001, and other mailing offices.

POSTMASTER
Please send address changes to
Communications of the ACM
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA

Printed in the USA.

Association for Computing Machinery

PLEASE RECYCLE THIS MAGAZINE

DOI:10.1145/3310359

Andrew A. Chien

Owning Computing's Environmental Impact

For decades, we have carried the conceit that "computing makes everything more efficient," so the impact of computing on the environment is a net positive.

But computing's unprecedented success has produced an explosion in its use, quantity, and direct environmental impact. It is time for the computing community to face up to computing's growing environmental impact—and take responsibility for it! And further, to undertake research, design, and operations to reduce this growing impact.[a]

The transition to SSDs and the consolidation of enterprise computing into efficient cloud datacenters has for a decade blunted the impact of growing computing use. But cloud computing's extraordinary scale (200TWh, $200B in 2017) and ICT's projected power growth (to 21% of global power consumption by 2030)[5] drive the rapid growth of the cloud's atmospheric carbon emissions.[3,6] Computing is the fastest-growing use of electric power in the developed world, and is driving the buildout of power generation and transmission in much of the developing world. If the world is to meet the Paris Accords goals for greenhouse-gas-emissions, computing must reduce its direct emissions.

Equally daunting is the rapid growth of waste from computing electronics, notably consumer products, smartphones, and the plethora of "smart devices" collectively termed the "Internet of Things." In 2016, e-waste reached 44.7 million metric tons per year, comparable to the size of the nine Pyramids at Giza, or 1.23 million 18-wheel trucks full of trash. This is an 8% increase from only two years earlier.[7] Of this massive quantity, only a fraction is collected and recycled, with the largest fraction simply dumped into landfills or incinerated. Any claims that computing is "good" for the environment, must reckon with this waste problem.

Some computing professionals believe that Moore's Law or Dennard scaling mitigates these problems. Far from it, they actually exacerbate it! Efficiency is not a solution, as 19th-century British Economist William Stanley Jevons noted in 1865, "efficiency increases consumption," a rule widely known as Jevon's Paradox.[4]

Carbon offsets are constructive, but not enough. As regions undertake ambitious 100% renewable fraction goals—San Diego (2035), California (2045), European Union (entire economy 2045)—offsets are of decreasing benefit. Real solutions must achieve direct matching and supply following.[2,8] Recycling programs are constructive, but less than 20% of e-waste is recycled—it is just not economic. Innovative approaches to capture or render benign e-waste are a critical need.

Recycling programs are constructive, but less than 20% of e-waste is recycled.

Computing technologies and systems must be designed and shaped for lower carbon and environmental impact. Here is a call to action to the computing community: Let's adopt goals equally ambitious to those of the climate community.

Let's create technologies and systems that in their manufacture, construction, and operation approach the goal of 100% carbon-free and neutral environmental impact!

Andrew A. Chien, EDITOR-IN-CHIEF

Andrew A. Chien is the William Eckhardt Distinguished Service Professor in the Department of Computer Science at the University of Chicago, Director of the CERES Center for Unstoppable Computing, and a Senior Scientist at Argonne National Laboratory.

References

1. Dreyfuss, E. How Google keeps its power hungry operations carbon neutral. *Wired* (Dec. 1, 2018).
2. Google White Paper. Moving toward 24x7 Carbon-Free Energy at Google Data Centers: Progress and Insights.
3. Greenpeace. Clicking Clean: Who is winning the race to build a green Internet?
4. Jevon, W. *The Coal Question*. Macmillan, 1865.
5. Jones, N. How to stop data centres from gobbling up the world's electricity. *Nature* (Sept. 12, 2018).
6. Shehabi, A. et al., United States Data Center Energy Usage Report. LBNL, June 2016.
7. United Nations University.The Global E-waste Monitor 2017 (Dec. 2017).
8. Yang, F. and Chien, A.A. ZCCloud: Exploring Wasted Green Power for High-Performance Computing, IPDPS, May 2016.

[a] Of course, I do not mean to imply that there have been no efforts to date—much to the contrary. But, rather to call for renewed and universal engagement on this agenda.

DOI:10.1145/3310251

Vinton G. Cerf

Ownership vs. Stewardship

I HAVE BEEN THINKING recently about the complementary roles of stewardship and ownership in the context of the Internet. A significant fraction of the physical infrastructure of the Internet and the equipment that animates the World Wide Web is privately owned. Some of these components are "owned" by governments and in some cases by cooperatives. The owners are usually motivated by the benefits of their ownership whether that is making a profit, fulfilling government obligations, or producing benefit for the group owners in the case of a cooperative. There are, of course, cases in which profit is not a motive but rather social benefit. Think of schools, churches, and libraries that provide access to the Internet freely. They are the owners of the facilities and operate them in part to fulfill their missions. So where does stewardship fit into this picture?

Organizations such as the Regional Internet Registries[a] (ARIN, RIPE, LACNIC, AFRINIC and APNIC) and the Internet Corporation for Assigned Names and Numbers[b] (ICANN) do not own the assets they administer. Their missions are to manage the assignment of these assets to parties who have the right to use these assets in exchange for fees paid to maintain these assignments. In the case of domain names, the registrars are often, but not always, for-profit entities that assist users to register and maintain a record of the assignment. The domain name registries may also be for-profit, non-profit, or even government operated in the case of the country-code Top Level Domains. Interestingly, in the case of domain names, the assets (that is, right to use) is created by the registrant who can invent new names not previously registered and register them for use. In the case of Internet address space, there is a finite amount of that space (IPv4 has 32 bits of address space, IPv6 has 128—*a lot*!) and the right to use is meted out by ICANN to the Regional Internet Registries, which is in turn typically meted out to Internet service providers who temporarily or even dynamically allocate addresses to end users for the purpose of their gaining access to the Internet and its many services.

The Regional Internet Registries and ICANN are stewards of IP address and domain name spaces. Their role is, to the best of their abilities, to provide fair access to these assets and to keep track of these assignments to ensure parties who do not have a registered right to use them cannot falsely claim or hijack them. While opinions may vary as to the success of these institutions in carrying out their missions, it strikes me as very interesting that the largely private sector ownership of the physical Internet (including the hosts and cloud datacenters) is ultimately dependent on the successful stewardship of a few key non-profit entities for the useful application of these physical assets. The Web would not exist in its present form without domain names. Domain names would not be useful if they could not be mapped into IP addresses so the servers of those domain names could be reached on the Internet.

Given the understandable economic motivations of for-profit institutions, the fact that the Internet and the World Wide Web are dependent on the stewardship of a key set of non-profit institutions strikes me as quite remarkable. It might easily have gone the other way. There is a current trend toward encouraging technology transfer from government-sponsored research into the private sector. I agree with the general premise that publicly funded research should find its way into the private sector where investment and hard work can produce economic gains, jobs, and useful products and services. It is nonetheless fascinating to me that one of the largest government-produced engines of economic growth and innovation is so deeply dependent on the stewardship of the people and institutions that administer Internet addresses and domain names.

The Internet Society[c] (ISOC) is another non-profit institution, which houses the Internet Architecture Board[d] (IAB), the Internet Engineering Task Force[e] (IETF), and the Internet Research Task Force[f] (IRTF). ISOC benefits directly from its wholly owned, non-profit subsidiary, the Public Interest Registry[g] (PIR), which is the steward of the .org top level domain. The Internet Society plays an active role in reminding us of the importance of stewardship and the need to heighten awareness that harmful uses of the Internet threaten its global connectedness and the safety of its users. We are all in debt to the stewards of the Internet. Long may they serve. C

[a] https://en.wikipedia.org/wiki/Regional_Internet_registry

[b] https://www.icann.org/

[c] http://www.isoc.org

[d] http://www.iab.org

[e] http://www.ietf.org

[f] http://www.irtf.org

[g] http://www.pir.org

Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. He served as ACM president from 2012–2014.

DOI:10.1145/3306448

Moshe Y. Vardi

Lost in Math?

WHEN I WAS 10 years old, my math teacher started a Math Club. It was not popular enough to last more than a few weeks, but that was long enough for me to learn about matrices and determinants. When I came home, my mother asked me how the club had been. "Beautiful," I answered. "Do you mean, 'interesting'?" she inquired. "No," I said, "Beautiful!" While some people find mathematics befuddling, others find it elegant and beautiful. The mathematician Paul Erdős often referred to "The Book" in which God keeps the most beautiful proofs of each mathematical theorem. The philosopher Bertrand Russell said, "Mathematics, rightly viewed, possesses not only truth, but supreme beauty." The beauty can be compelling; something so beautiful must be true!

But the seductive power of mathematical beauty has come under criticism lately. In *Lost in Math*, a book published earlier this year, the theoretical physicist Sabine Hossenfelder asserts that mathematical elegance led physics astray. Specifically, she argues that several branches of physics, including string theory and quantum gravity, have come to view mathematical beauty as a truth criterion, in the absence of experimental data to confirm or refute these theories. The theoretical physics community, she argues, is falling victim to group thinking and cognitive bias, seduced by mathematical beauty. About 10 years ago, in the wake of the 2008 financial crisis, the Nobel Laureate economist Paul Krugman made the same point with respect to economics and mathematics in an influential article titled "How Did Economists Get It So Wrong?" His main answer was: mistaking mathematical beauty for truth. "As I see it," wrote Krugman, "the economics profession went astray because economists, as a group, mistook beauty, clad in impressive-looking mathematics, for truth."

So both physics and economics have, arguably, been lost in math. What about computer science? Specifically, what about theoretical computer science (TCS)? TCS is surely blessed with mathematical beauty. As a graduate student a long time ago, it was mathematical beauty that attracted me to TCS, and continued to lead my research for many years. I find computational complexity theory (or complexity theory, for short), with its theorems (for example, the time-hierarchy and space-hierarchy theorems) and its open questions (for example, *P* vs *NP*), to be hauntingly beautiful. Beauty, yes; but what about truth?

Physical theories describe the physical world, and by their "truth" we refer to the fidelity in which they describe this world. Economic theories describe economic systems, but by their "truth" we refer not only to the fidelity in which they describe such systems but also to the quality of the guidance they offer to business people and policymakers. I believe complexity theory is similar to economic theories in that respect. It should not also provide a theoretical framework in which we can study the performance of algorithms, but it should also offer sound guidance to algorithm designers and system developers who use algorithms. So how good a theory is complexity theory from that perspective?

It is clear what it means to measure the performance of a specific algorithm *A* on a specific problem instance *I*. But complexity theory aims at describing the performance of *A* over the space of *all* problem instances and it does so by abstracting away from individual problem instances. The typical way in which we do this abstraction is by considering *all* problem instances of length *n*, and asking for upper and lower bounds on algorithmic performance (usually in terms of time and memory utilization) as a function of *n*. This approach is referred to as the *worst-case* approach, as it focused on the most challenging problem instance of each length *n*. If the upper bound that we can prove is one of a slow-growing function, for example, *cn log n*, for a small constant *c*, then we have a guarantee of good performance on *all* problem instances. But, in general, most upper and lower bound are much less useful. For example, an exponential lower bound just says some problem instances are hard, but says nothing about the practical significance of such instances.

In previous columns I have discussed this gap between theory and practice in specific settings. As I pointed out, program termination may be unsolvable in theory but solvable in practice,[a] while Boolean satisfiability may be intractable in theory but tractable in practice.[b] In both cases, the worst-case approach is simply too pessimistic and tells us too little about algorithmic performance in practice. Beauty does not necessarily entail truth. Going beyond worst-case complexity is a key challenge in complexity theory and is the subject of much current research. (See Tim Roughgarden's Review Article on p. 88).

Follow me on Facebook, Google+, and Twitter.

[a] https://cacm.acm.org/magazines/2011/7/109895-solving-the-unsolvable/fulltext

[b] https://cacm.acm.org/magazines/2014/3/172516-boolean-satisfiability/fulltext

Moshe Y. Vardi (vardi@cs.rice.edu) is the Karen Ostrum George Distinguished Service Professor in Computational Engineering and Director of the Ken Kennedy Institute for Information Technology at Rice University, Houston, TX, USA. He is the former Editor-in-Chief of *Communications*.

BLOG@CACM

The *Communications* Web site, http://cacm.acm.org, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.

twitter

Follow us on Twitter at http://twitter.com/blogCACM

DOI:10.1145/3303708 http://cacm.acm.org/blogs/blog-cacm

Smoothing the Path to Computing; Pondering Uses for Big Data

Members of the Computing Research Association suggest ways to broaden participation in computer science, while Saurabh Bagchi looks at use cases for big data.

Mary Hall, Richard Ladner, Diane Levitt, Manuel Pérez-Quiñones

Broadening Participation in Computing Is Easier Than You Think

https://cacm.acm.org/blogs/blog-cacm/233339-broadening-participation-in-computing-is-easier-than-you-think/fulltext

December 11, 2018

The U.S. National Science Foundation (NSF) recently introduced new requirements for the Computer and Information Science and Engineering (CISE) Directorate programs, whereby some funded projects must include a Broadening Participation in Computing (BPC) Plan. To facilitate this transition, the Computing Research Association (CRA) is launching a resource portal called BPCnet (https://bpcnet.org), which is being funded by NSF to connect organizations that provide BPC programs with computing departments and NSF grant proposers. These changes reflect a recognition that any significant impact on the diversity of the field will benefit greatly from *engaging the entire academic computing research community*. Many universities will respond by expanding their broadening participation efforts to include students from groups who are underrepresented in computing, including women, underrepresented minorities, and students with disabilities (URMD). Here we list 10 small steps departments can do toward this goal.

1. **Organize departmental BPC efforts at your university:** Create a sign-up list of diversity activities, and incentivize faculty to participate. Create a departmental strategic plan for broadening participation that faculty can support and amplify in their funded NSF CISE proposals. Consider how to leverage BPCnet providers as part of your departmental plan.

2. **Optics matter:** Include pictures of URMD students in websites and printed materials. Artwork, examples in class, etc., should appeal to all students and not reinforce stereotypes. The same goes for examples you present in class. If you think they fail to be inclusive, they probably are.

3. **Make departmental infrastructure accessible, inclusive, internationalized:** Provide accessible classrooms, labs, offices, websites, videos, etc. Use international alphabets for student names. Ask students for their preferred pronouns.

4. **Measure and track:** Analyze your enrollment, demographics, etc., regularly to identify problem areas and track changes, on your own, or with the CRA Data Buddies.

5. **Create a community for URMD students:** Sponsor student organizations, and send students to Grace Hopper, Tapia, and other celebrations of diversity in computing.

6. **Recruit URMD teaching assistants, professors, advisors:** Representation matters. Students value seeing someone who looks like them being successful in their field.

7. **Promote undergraduate research:** Work with women and URMD students in undergraduate research projects, such as through CRA's CREU and DREU.

8. **Create curriculum enhancements that appeal to diverse students:** Create introductory courses that assume no computing background, CS+X degree programs, service-learning, and accessibility electives.

9. **Develop the K–12 pipeline:** Work with K–12 teachers (CSTA) and improve state curricula (ECEP) to advance K–12 computing education.

10. **Engage the community to stimulate computing interest and skills:** Organize rigorous and joyful outreach

events that bring diverse K–12 students and their families onto your campus.

From "Increasing Diversity In Computing Is Easier Than You Think: Some Small Steps That Can Make A Big Difference," panel, 2018 CRA Conference at Snowbird, UT.

Saurabh Bagchi Short Take: Big Data and IoT in Practice

https://cacm.acm.org/blogs/blog-cacm/233312-short-take-big-data-and-iot-in-practice/fulltext

December 10, 2018

Beyond the tremendous level of activity around big data (data science, machine learning, data analytics … take your pick of terms) in research circles, I wanted to peek into some of the use cases for its adoption in the industries that deal with physical things, as opposed to digital objects, and draw some inferences about what conditions help adoption of the research we do in academic circles.

What's Driving the Convergence?

The convergence of Internet of Things (IoT) and big data is not surprising at all. Industries with lots of small assets (think pallets on a factory floor) or several large assets (think jet engines) have been putting many sensors on them. These sensors generate unending streams of data, thus satisfying two of the three V's of big data right there: velocity and volume. Next time you are on a plane and are lucky to be next to the wings, look underneath the wings and you will see an engine — if it is Rolls Royce or GE, it may even have been designed or manufactured in our backyard in Indiana. Engines like these are generating 10 GB/s of data (http://bit.ly/2LTsMjy) that is being fed back in real time to some onboard storage or more futuristically streamed to the vendor's private cloud. This is one piece of the IoT-big data puzzle, the data generation and transmission. This is the more mature part of the adoption story (http://bit.ly/2SzWTz3). The more evolving part of the big data story is the analysis of all this data to make actionable decisions, and that, too, in double-quick time.

Use Cases for Collecting Big Data

The second part of this story is in the analysis of all this data to generate actionable information. Talking to my industrial colleagues, there are five major use cases for such analysis:

1. **Predictive maintenance/downtime minimization:** Know when a component is going to fail before it fails, and swap it out or fix it.

2. **Inventory tracking/loss prevention:** Many industries of physical analog things have lots of moving parts; again, think of pallets being moved around. They want to track where a moving part is now and where all it has been.

3. **Asset utilization:** Get the right component to the right place at the right time so that it can be used more often.

4. **Energy usage optimization:** Self-explanatory, and increasingly important as the moral and dollar imperatives of reducing energy usage become more pressing.

5. **Demand forecasting/capacity planning:** Self-explanatory, but firms seem to be getting better at this at shorter time scales. Way back in 1969, the U.S. Federal Aviation Administration (FAA) was predicting air traffic demands on an annual basis (http://bit.ly/2BWj5fv); now think of predicting the demand for the World Cup soccer jerseys depending on which country is doing how well on a daily basis (http://bit.ly/2R3IcHL).

Factors Helping Adoption of Academic Research

Academia has been agog about this field of big data for, well, … seems like forever. We academics thirst for real use cases and real data and this field exemplifies this more than most. We need to be able to demonstrate our algorithm and its instantiation in a working software system delivers value to some application domain. How do we do that? There is a lot of pavement pounding and trying to convince our industrial colleagues. Again talking to a spectrum, some factors seem to recur frequently. These are not universal across application domains, but they are not one-off, either.

1. **Horizontal and vertical.** There is a core of horizontal algorithmic rigor that cuts across the specifics of the application, but this is combined quite intricately with application-specific design choices. We can snarkily call them "hacks," but they are supremely important pieces of the puzzle. This means we cannot build the horizontal and throw it across the fence, but rather have to go the distance of understanding the application context and the vertical.

2. **Interpretability.** While ardent devotees at the altar of big data are willing to accept the output of an algorithm like the Oracle of Delphi, many of my industrial colleagues in the business of building physical objects small or large are cagey about such blind faith. Thus, our algorithms must provide some insights or knobs to play "what-if" scenarios. This sometimes runs at odds with building superpowerful models and algorithms, but it is our dictate from the real world to make smart trade-offs.

3. **Streaming data and warehouse data.** My colleagues seem to want the yin and the yang on the same platform. The data analytics routine should be capable of handling data as it streams past, as well as old data from years of operation that is sitting in a musty digital warehouse. This speaks to the need to extract value from the wealth of historical data, as well as making agile decisions on the streams of data being generated now.

4. **Unsupervised learning.** This is entering technical-jargonland, but basically this means we do *not* want to have to recruit armies of people to label data before we can let any algorithm loose on the data. That takes time, effort, legal wrangling, and we are never completely sure of the quality of labeling. So we would, whenever we can, use unsupervised learning, which does not rely on a deluge of labeled data.

Conclusion

The domains of big data and IoT are destined to mutually propel each other. The former makes the latter appear smarter, even when the IoT system is built out of lots of small, dumb devices. The latter provides the former with fruitful, challenging technical problems. Big data algorithms here have to become small, run with a small footprint, a gentle giant in the land of many, many devices.

Mary Hall is a professor in the School of Computing at the University of Utah, and a member of the Computing Research Association Board. **Richard Ladner** is professor emeritus in the Paul G. Allen School of Computer Science & Engineering at the University of Washington. **Diane Levitt** is the senior director of K–12 Education at Cornell Tech. **Manuel A. Pérez Quiñones** is associate dean of the College of Computing and Informatics at the University of North Carolina at Charlotte, and professor in the Department of Software and Information Systems. **Saurabh Bagchi** is a professor of electrical and computer engineering, and of computer science, at Purdue University, where he leads a university-wide center on resilience called CRISP.

N news

Science | DOI:10.1145/3303710

Logan Kugler

Building a Better Battery

How researchers are improving energy storage devices for power generated from renewable sources like solar and wind.

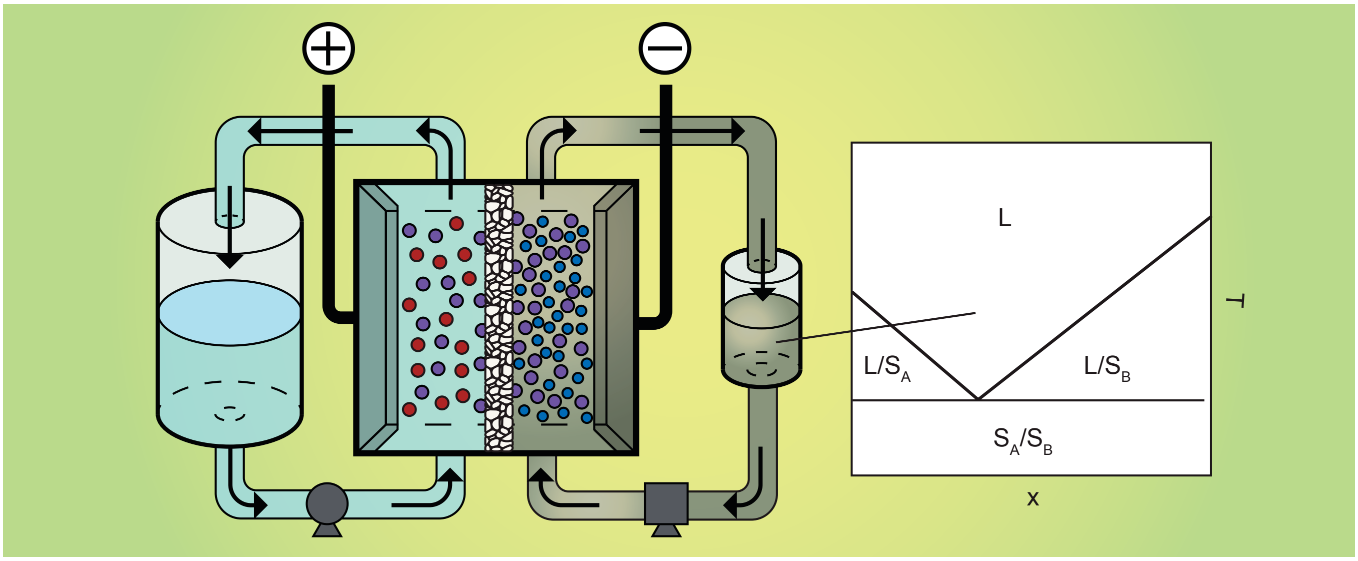
THE WORLD'S RENEWABLE power capacity—the amount of energy that can be produced from sources comprising hydropower, wind, solar, bioenergy, and geothermal—has doubled in the last decade. From 2008 to 2017, it went from 1,060 gigawatts to 2,179 gigawatts, according to research from the International Renewable Energy Agency (IRENA), an intergovernmental organization.

Over that period, world renewable energy production has also increased, from over 3.7 million gigawatt hours to almost 5.9 million gigawatt hours.

To be clear, capacity, according to the U.S. Department of Energy, is the "maximum output of electricity that a generator can produce under ideal conditions," whereas generation is what actually gets produced.

In both cases, much of the growth, according to data from IRENA, comes from wind and solar sources. This is good news for clean energy advocates who hope to see our world less reliant on polluting fossil fuels that contribute to climate change.

Renewable power sources provide "fuel" that is free and clean. This is highly attractive to businesses that generate electricity, and those that run on electricity. Yet sources like wind and solar are also intermittent; after all, the wind stops blowing at times,



A flow battery in which energy is stored in a sodium-potassium alloy.

IMAGE CREDIT: ANTONIO BACLIG

and the sun stops shining at night. In contrast, power generated from coal and natural gas is consistent, and able to be ramped up and down quickly to meet demand.

This presents a conundrum for clean energy advocates. How do you make renewable energy not only cheap enough, but reliable enough, to compete with fossil fuels?

The answer may lie in the types of energy storage used to capture the power generated from these renewable sources. Better batteries could make it possible for utility companies to leverage renewable energy sources at scale as primary power sources, rather than as clean backups to traditional dirty fuels.

Powering the Grid

The power grids that charge and recharge modern life are complicated beasts, comprised of many different companies and technologies that make it possible to generate power, then route that power to homes and businesses.

These grids operate on a basic principle: once energy is generated, it is sent out for consumption. If it is not being consumed immediately, it needs to be stored in some fashion so it can be released back into the grid when end users are ready to consume it.

It sounds easy in theory, but it quickly gets complicated in practice.

When businesses and homes need power, they typically need it immediately. This means demand for grid power can fluctuate a lot, and it can fluctuate quickly. Power supply must match demand as adequately as possible, on time and on budget. Failure to meet demand sufficiently and on time results in power failures or blackouts. Failure to meet demand on-budget means a power provider eventually goes out of business.

This is where the benefits of fossil fuels become evident. Fossil fuel-powered energy plants generate consistent power, since humans, not nature, control their fuel sources, and the different fossil fuel power generation technologies (such as coal, oil, and natural gas) can be scaled up or down to meet demand fluctuations. Despite being dirty (basically, the fuels are burned to heat water to turn turbines that generate electricity, releasing carbon dioxide and other contaminants into the atmosphere in the process), they are (at the moment) reliable.

"Incumbent technologies are still on a steep cost-down curve, which is challenging for new technologies to compete with," says Stanford University's William Chueh.

Renewable energy sources, on the other hand, are clean and their fuel is free and abundant. However, the power generated by water, wind, and solar sources must be stored somewhere after it is generated, since it is intermittent.

The top method utilized in the U.S. for renewable energy storage is pumped-storage hydroelectric, which provides 95% of grid-scale electricity storage in the country, according to the U.S. Department of Energy. Pumped-storage hydro utilizes multiple reservoirs to store and release electricity in a highly efficient and responsive manner, allowing power grids to react quickly to fluctuations in demand.

According to electric power holding company Duke Energy, "Pumped-storage hydro plants store and generate energy by moving water between two reservoirs at different elevations." When demand is low, "Excess energy is used to pump water to an upper reservoir." When there's high demand, the water is released from the upper reservoir to generate power.

However, building these storage systems requires massive time and resource investments, and the location of these storage systems is highly dependent on geography.

This is where battery technology comes in. On the grid, the right battery technology may be able to store and release energy at far less cost than pumped-storage hydro. Some battery developments are also enabling the transition to electric vehicles, further reducing reliance on non-renewable fossil fuels.

One of the main battery types currently used for energy storage is the lithium-ion battery. Lithium-ion batteries power the handheld tech gadgets we use every day, including smartphones. They also provide energy storage for electric cars, and extremely large lithium-ion batteries are increasingly being used in power grids to store energy from renewable sources.

Lithium-ion batteries are inexpensive and energy-dense compared to batteries made with other materials. They also degrade relatively slowly, losing just a fraction of their power after each use.

"Over the past decade, we have seen a tremendous cost reduction of lithium-ion battery technology, by approximately 10 times," says William Chueh, an assistant professor of Materials Science and Engineering at Stanford University working on renewable energy storage technologies. "This has been responsible for the boom in electric vehicles and for storing intermittent solar and wind electricity."

However, lithium-ion batteries have one big problem: they still are not priced competitively enough to be used at scale on grids to store energy from renewable sources.

"To realize a complete penetration of batteries for storing intermittent renewable electricity, the cost needs to decrease by another order of magnitude, and the scalability needs to be greatly improved," says Chueh.

Those developments are unlikely, says Ben Schiltz, head of Energy Storage Communications at the U.S. Department of Energy's Argonne National Laboratory, which has multiple energy storage research projects in progress.

"Today, the industry continues to make incremental improvements to lithium-ion batteries. However, we are reaching the theoretical limit of what can be done with these batteries," Schiltz says. "Developing safe new battery technologies with higher energy capacity, lower cost, and longer life, that can also be charged and discharged fast, is very challenging. Oftentimes you can improve one [factor] at the expense of others."

This situation has researchers hunting for alternative technologies to store energy from renewable sources.

Alternative Energy Storage

There are many alternatives when it comes to grid energy storage, says Antonio Baclig, a renewable energy storage researcher at Stanford and a member of Chueh's team.

"Lithium-ion is now the front-runner for grid storage batteries, but lead-acid batteries are a low-cost alternative, and flow batteries are continuing to improve," he says. Lead-acid batteries are solid batteries used in automotive applications. Rechargable flow batteries, however, use liquids instead of solids to conduct electricity, which may give researchers more options to find chemical combinations that dramatically improve efficiencies and reduce costs.

Lead-acid batteries have the "largest market share for rechargeable batteries," according to research published in *The Journal of Energy Storage* by Geoffrey May, Alistair Davidson, and Boris Monahov. Much of the market for these batteries is in traditional motor vehicles—standard car batteries.

While lead-acid batteries are relatively inexpensive and widely available, they do have some drawbacks: for one, they are heavy, which makes them less than ideal for electric cars (Tesla's vehicles, for example, use lithium-ion batteries.)

The real advantage of lead-acid over lithium-ion has historically been cost, but that is changing. Research by Joe O'Connor, manager of application engineering at battery technology company Farasis Energy Inc., showed that while individual lead-acid batteries are cheaper to buy than lithium-ion batteries, the total lifecycle cost for off-grid lithium-ion batteries is reaching parity with that of lead-acid batteries.

Lithium-ion batteries require little maintenance and are more resilient to irregular discharging than lead-acid batteries, according to O'Connor.

While lead-acid and lithium-ion technologies duke it out over marginal cost reductions and efficiency gains, advances and improvements in flow batteries may just be getting started.

Chueh, Baclig, and a team of researchers have developed a new type of flow battery that could eventually be a low-cost, high-power alternative to existing energy storage methods.

According to Stanford, "The group found a suitable ceramic membrane made of potassium and aluminum oxide to keep the negative and positive materials separate while allowing current to flow." The membrane "doubled the maximum voltage of conventional flow batteries, and the prototype remained stable for thousands of hours of operation."

With those advances, says Chueh, "We aim to simultaneously achieve high energy density, lifetime, and reduced cost."

However, this new flow battery is still in the prototype phase. While promising, it will not be mass-produced any time soon.

Another new battery type that is commercially farther along in scaling the storage of renewable energy is the zinc-air battery, a metal-air battery powered by oxidizing zinc with oxygen from the atmosphere. These batteries have high energy densities, and are relatively inexpensive to produce. Late last year, energy storage technology company NantEnergy and its billionaire founder Patrick Soon-Shiong announced they had developed a battery that uses zinc and air to store renewable energy.

This zinc-air battery has been tested "in Africa and Asia, as well as cellphone towers in the United States for the last six years, without any backup from utilities or the electric grid," according to *The New York Times*. The company claims the new battery can store and release electricity at a cost of less than $100 per kilowatt-hour (kWh). In comparison, Elon Musk told shareholders that Tesla was working to get to that price point for its lithium-ion battery cells by the end of last year.

The $100/kWh mark is seen in the energy storage community as a tipping point for widescale adoption of electric cars, according to Bloomberg.

Despite these advances, commercially viable and scaleable grid-ready alternatives to lithium-ion batteries remain to be seen.

"Incumbent technologies are still on a steep cost-down curve, which is challenging for new technologies to compete with," says Chueh. However, Chueh is confident that, given a long-enough timeline, alternative grid batteries will be part of the answer to scaling renewable power storage.

"Batteries used at the grid scale will look more like a chemical plant, rather than the batteries used in electric vehicles, drones, and robots today," Chueh says. C

Further Reading

May, G.
Lead batteries for utility energy storage: A review, *Journal of Energy Storage*, February 2018, https://www.sciencedirect.com/science/article/pii/S2352152X17304437

O'Connor, J.
Battery Showdown: Lead-Acid vs. Lithium-Ion, *Medium*, Jan. 23, 2017, https://medium.com/solar-microgrid/battery-showdown-lead-acid-vs-lithium-ion-1d37a1998287

Penn, I.
Cheaper Battery Is Unveiled as a Step to a Carbon-Free Grid, *The New York Times*, Sept. 26, 2018 https://nyti.ms/2MiFwAj

Whiteman, A.
Renewable Energy Statistics 2018, *International Renewable Energy Agency*, July 2018 http://www.irena.org/publications/2018/Jul/Renewable-Energy-Statistics-2018

Uria-Martinez, R.
2017 Hydropower Market Report, *U.S. Department of Energy*, April 2018 http://bit.ly/2TWJUaN

Pumped-Storage Hydro Plants, *Duke Energy* http://bit.ly/2MfvhN6

What's the Difference Between Installed Capacity and Electricity Generation?, *U.S. Department of Energy*, Aug. 7, 2017 http://bit.ly/2MhLrWv

Logan Kugler is a freelance technology writer based in Tampa, FL, USA. He has written for over 60 major publications.

Technology | DOI:10.1145/3303851

Esther Shein

Exoskeletons Today

Wearable mobile machines integrate people and machines to assist the movement-impaired, and amplify the capabilities of industrial and defense workers while protecting them from injury.

Millions of people suffer from the effects of spinal cord injuries and strokes that have left them paralyzed. Millions more suffer from back pain, which makes movement painful. Exoskeletons are helping the paralyzed to walk again, enabling soldiers to carry heavy loads, and workers to lift heavy objects with greater ease.

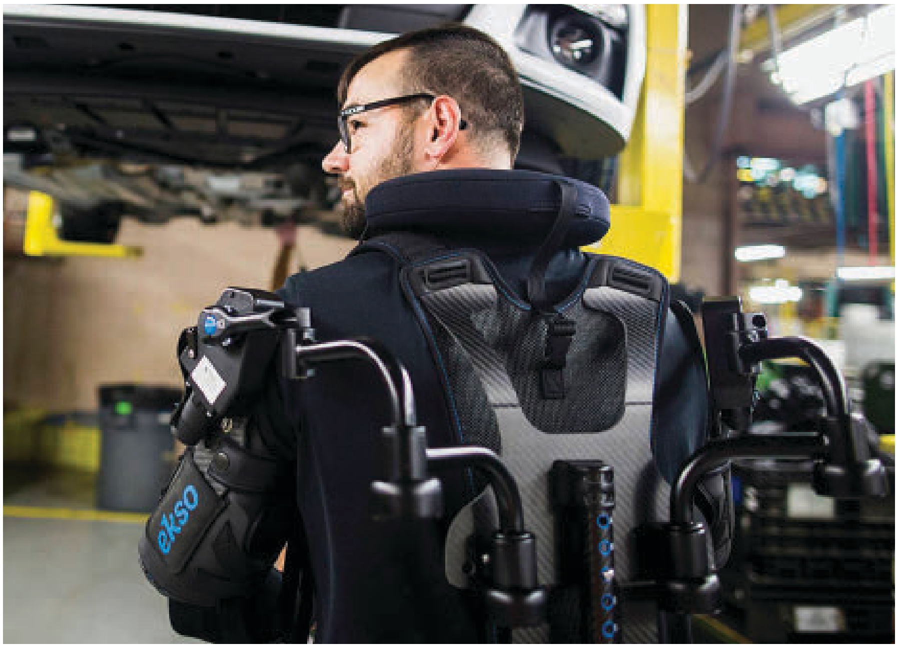
An exoskeleton is a mechanical device or soft material worn by a patient/operator, whose structure mirrors the skeletal structure of the operator's limbs (joints, muscles, etc.). The structure works in tandem with the person wearing it, and it is utilized to amplify their capabilities, serving as an assistive device, haptic controller, or for rehabilitation purposes, says Rian Whitton, an analyst at technology market intelligence firm ABI Research. The firm is forecasting 150,618 exoskeleton shipments in 2028 and $2.9 billion in revenue in 2028, up from $104 million in 2018.

The technology has been around since the 1960s; during the Cold War, however, the focus was mainly on research and exploration, Whitton says. Research and development activity picked up again in 2017, he says, when exoskeletons began to gain regulatory approval and were popularized in the health market.

"You have this increased market due to the number of veterans with spinal cord injuries resulting from conflicts post-9/11, and the prosthetics market and robotic limbs market was accelerated because of that now,'' Whitton says. "So you could argue the military … had a stimulating effect."

That said, exoskeletons are still in a "very nascent state,'' Whitton adds.

Hermano Krebs, the self-described "father of rehabilitative robotics" and principal research scientist in the Mechanical Engineering Department of the Massachusetts Institute of Technology (MIT), agrees. Until the 1970s and 1980s, he recalls, the perception was that the brain was hardwired, and there was not much that could be done for a person who suffered a stroke. Today, the concept of neuroplasticity, the brain's ability to form new neural connections, has shown that a person's brain functioning can grow stronger after suffering a stroke, says Krebs.



A Ford worker wearing the EksoVest, which provides lift assistance for repetitive overhead tasks.

IMAGE COURTESY OF EKSO BIONICS AND FORD

When Krebs began his research, his goal was to "create tools to help a clinician take advantage of the nurture that we could offer over the nature."

Applying robotics using concepts from neuroscience is helping people recover what they lost, he says. "It's not a cure, but it improves care … This is the direction of rehabilitative technology."

There are different types of robotics. In one scenario, a robot might be mounted on a wheelchair and feed a person by bringing a spoon to their face, which is an assistive application. Alternatively, an exoskeleton robot could be mounted around the person's arm to help them bring the spoon to their own face.

"Both have value, and it depends on what the application is,'' Krebs says. "Now you can think of [exoskeletons] … not just in rehabilitative medicine, but in the aging of the population or for workers in factories, depending on what the goal is."

Krebs and a team at MIT created a plastic exoskeleton known as Anklebot during 2003–2004. Anklebot is mounted around a person's ankle to help them after a stroke, since many stroke victims cannot lift their ankles to clear the floor when walking, he says. The Anklebot is designed to help the person propel him/herself forward and clear the floor by lifting the ankles so they don't fall.

Today, Krebs says, many U.S.-based companies have commercialized the Anklebot to treat patients, while several others have developed their own technologies for helping post-stroke patients lift their ankles. One is Japanese-based Yaskawa, which also has an ankle robot to help stroke patients. Krebs also started a new company called 4Motion Robotics, which will be designing an anklebot and other exoskeletal products.

Increased use of exoskeletons is also being driven by re-shoring, the bringing back of manufacturing jobs to the U.S. from offshore, notes Whitton. Exoskeletons can be beneficial in helping workers avoid injuries and stay on the job longer because they "amplify human performance," which results in a productivity gain.

"There is an acute labor shortage in industrial jobs here; there is a low participation rate and a lot of people are feeling the pinch and [companies are] struggling to grow and raise productivity due to a lack of workers and also an aging workforce." Already, people are retiring later, he noted, "so in a sense, this is a way of extending human life and extending their time in the labor force."

Couple those factors with injuries on construction sites and mining operations, and you can understand why exoskeletons are being eyed as a way to extend the life of the worker, because they can amplify performance.

Take, for example, a Milwaukee Grinder, a power tool mainly used to grind metal in discrete manufacturing, and a piece of equipment that can weigh 15 lbs. or more, Whitton says. An exoskeleton with a third zero-gravity arm could pick it up without requiring any exertion on the part of a worker, he says.

That appeals to home improvement chain Lowe's, which tested exosuits in April 2017 at its Christiansburg, VA, location, in partnership with Virginia Polytechnic Institute and State University (Virginia Tech). "We gathered feedback from the test and are now using the data to help define the next phase of the program,'' says a Lowe's spokesman, who declined to provide further details.

Ford Motor Co. is also testing at one of its assembly plants a wearable exoskeleton called EksoVest to help reduce shoulder injury, which is an issue in assembly line work.

The flip side of exoskeletons, however, is that it could take a while for a company to recoup its initial investment because they are extremely expensive right now, notes Whitton. An upper-body exoskeleton designed to amplify human performance runs about $30,000, he says, "but the price is coming down, and eventually these technologies will be commoditized." Additionally, he anticipates the cost of an exoskeleton will shift from hardware to software and robotics as a service with a monthly subscription model; this, he believes, will lower the barrier to adoption.

Exoskeletons can be beneficial in helping workers avoid injuries and stay on the job longer because they "amplify human performance," which results in a productivity gain.

Jerryll Noorden, a Connecticut-based real estate investor, is also bullish on exoskeletons. Prior to real estate, Noorden was the mechanical lead engineer on the National Aeronautics and Space Administration (NASA) X1 Exoskeleton, as well as working at the Florida Institute for Human & Machine Cognition (IMHC), a research institution of the State University System of Florida. Noorden believes so much in the viability of exoskeletons that he says he is planning to donate some of the proceeds from his business to exoskeleton research and development.

While working at IHMC in 2007, Noorden and others developed a prototype for an exoskeleton, which he says was "not very successful" because a motor was required for each joint to introduce movement. "The stronger the motor, the bigger it has to be, and that is the huge issue with motors right now,'' Noorden explains. "The more power, the bigger the motor, and of course, it becomes heavier."

ACM Member News

HELPING PEOPLE THROUGH INFORMATION CENTRICITY

"Many of us in research are very fortunate because we get to do what we love," says K.K. Ramakrishnan, professor of computer science and engineering at the University of California, Riverside (UC Riverside).

Ramakrishnan's research is focused on network architecture, protocols, and systems. He is author of more 250 papers, and has 170 patents issued in his name.

He earned his bachelor's degree in electronic engineering from Bangalore University in India, and his master's degree in engineering from the Indian Institute of Science (IISc), also in Bangalore.

While at IISc, Ramakrishnan used a computer for the first time, and was attracted to learn more about the computer arena because it was so intuitive. Computer science was a natural fit for him, and he earned his Ph.D. in computer science from the University of Maryland.

Ramakrishnan began his professional career at Digital Equipment Corporation (DEC) in 1983. He left DEC in 1984 to take a position with AT&T, where he worked through 2013, the year he joined the faculty of UC Riverside.

Currently, Ramakrishnan's research is focused on two areas. The first is using information centricity for disaster management, to deliver more timely, relevant, useful information to aid first responders. The second is to continue making networks more flexible and agile through software-based functionality.

Ramakrishnan observes that networks have become ubiquitous, and that much of our life is centered around a connected world. As a result, he says, the future lies in making these networks "more robust, convenient, and secure."

—John Delaney

The IHMC team came up with a second prototype, called Mina, in the 2008–2009 timeframe, which was much more successful, according to Noorden. At that point, he was contracted by IHMC to NASA, which he says was interested in the technology to help astronauts exercise in space.

However, after dealing with the fallout of budget cuts at NASA, Noorden decided to go out on his own and start a real estate investing business, "so that I could continue doing research without having to ask the government for funding."

In the meantime, several things need to happen for exoskeletons to improve, observers say. Like Noorden, Krebs says, "We're still not only trying to hide the hardware, but make it lighter. Many of the actuators and motors are still bulky, and it's not easy to reduce the weight of the devices." If the weight is reduced, "many times we don't have the behavior we need," meaning when the motor becomes smaller, it becomes more difficult for a person to move. "Ultimately, you want to devise something transparent that will help move you and assist you but won't hold you back. But we're not there yet."

Not only is the goal to make a exoskeleton device lighter, but also for it to behave in a way that people feel good about it, so it does not hold them back or prevent them from doing a particular movement. "This balance is what makes engineering difficult," Krebs says.

Yet, Krebs believes within 10 years the technology will have come far enough that it will be "far more available in multiple settings."

He also thinks exoskeletons will be front and center at the Tokyo Summer Olympics games in 2020. Toyota's Partner Robot division has developed a device to exercise the knee of a patient, and many other Japanese-based companies also are working on exoskeleton rehabilitation devices, he says. "I think the [Tokyo] Paralympics will be more interesting than the regular Olympics," he says, "because there will be lots of demonstrations of technology to help people move," due to the number of Japanese companies that are working on exoskeletal technology now.

> **"Ultimately, you want to devise something transparent that will help move you and assist you but won't hold you back. But we're not there yet."**

In healthcare, the main goal is to make exoskeletal devices "essentially disappear" and become "soft exoskeletals." Many people are working right now on incorporating such systems into clothing, Krebs says, such as a pair of pants with wires or cables that would be able to assist a person's mobility. One is an Israeli company called ReWalk Robotics, which has partnered with The Wyss Institute at Harvard University to make assistive exosuits devices for people with lower-limb disabilities.

Whitton thinks that as a result of the cost and "complexity of the health supply chain," due to a strict regulatory environment and the need to tailor an exoskeleton to an individual, healthcare will not be the biggest market for exoskeletons. He anticipates greater adoption in industries like manufacturing, mining, and defense.

"In many cases, a robot isn't sufficiently adaptable or dexterous to perform a wide number of tasks,'' Whitton explains, whereas industrial and mobile robots and exoskeletons are aimed at assisting human workers to enable them to do more, and more easily.

Whitton also foresees the systems becoming lighter and more dexterous and expects them to be deployed like Internet of Things (IoT) devices with artificial intelligence (AI) capabilities such as data analytics, to monitor worker performance and measure when a worker might be most at risk for injury. Exoskeletons are going to be laden with sensors and will be connected with other wearables like mobile control panels, he says. "While current systems are somewhat rudimentary, the aim is to track historical data about worker body positioning and to track [an] exosuit's location through the workspace."

Significant progress needs to be made on the social side, too. "There needs to be a greater sense of their transformative potential and the potential to get sufficient ROI in a sufficient amount of time,'' Whitton adds. Exoskeletons are "a capital-intensive technology and we don't know when the ROI will be. I think that will change over time as they are deployed by bigger companies." C

Further Reading

Rupal, B.S., Rafique, S., Singla, A., Singla, E., Isaksson, M., and Virk, G.S.
Lower-limb exoskeletons: Research trends and regulatory guidelines in medical and non-medical applications, *International Journal of Advanced Robotic Systems*, 2017.

Luo, J., Pan, B., and Fu, Y.
**Experiment research of human lower extremity exoskeleton robot, *Proceedings of the 32nd Chinese Control Conference*, 2013
https://ieeexplore.ieee.org/document/6640490/authors#authors**

Baldovino, R., and Jamisola, R.
**A Study on the State of Powered-Exoskeleton Design for Lower Extremities, *5th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management 2009* (HNICEM 2009)
http://bit.ly/2HhaWIl**

Neuhaus, P.D., Noorden, J.H., Craig, T.J., Torres, T., Kirschbaum, J., and Pratt, J.E.
**Design and Evaluation of Mina a Robotic Orthosis for Paraplegics, *Proceedings of the 2011 International Conference on Rehabilitation Robotics (ICORR 2011)*, Zurich, Switzerland
http://bit.ly/2Ct9E7H**

Kwa, H.K., Noorden, J.H., Missel, M., Craig, T., Pratt, J.E., and Neuhaus, P.D.
**Development of the IHMC Mobility Assist Exoskeleton, *Proceedings of the 2009 IEEE International Conference on Robotics and Automation, (ICRA '09)*, Kobe, Japan
http://bit.ly/2Fzn2uJ**

Gui,L., Yang, Z., Yang, X, Gu, W., and Zhang, Y.
**Design and Control Technique Research of Exoskeleton Suit, *IEEE International Conference on Automation and Logistics*, 2007
https://ieeexplore.ieee.org/document/4338624/authors#authors**

Esther Shein is a freelance technology and business writer based in the Boston area.

Society | DOI:10.1145/3303847

Keith Kirkpatrick

Electronics Need Rare Earths

Demand is expected to spike over the next few years, leading to higher prices and international trade issues.

RARE-EARTH ELEMENTS are special minerals used in a wide variety of consumer and industrial products. Though they have exotic-sounding names, such as neodymium, scandium, and dysprosium, they are abundant right here on Earth. They are considered rare, however, because they appear in very small concentrations.

In addition, the process used to separate them from the rocks in which they occur is extremely difficult, because the elements have the same ionic charge and are similar in size. Typical separation and purification processes often require thousands of extraction and purification stages to be carried out. As such, there is a significant premium attached to these materials, and several market and geopolitical forces may cause them to escalate in value.

Rare earths are metallic elements, and therefore contain unique properties, including high heat resistance, strong magnetism, high electrical conductivity, and high luster. These specific properties make them well suited for use in a variety of products, including cellphones, batteries, loudspeakers, lights, magnets, and even wind turbines. In addition, they are often key elements used in the creation of components used in everyday objects, such as light-emitting diodes (LEDs), fiber optics, compact fluorescent lights, and are used as catalysts, phosphors, and polishing compounds for air pollution control, illuminated screens on electronic devices, and the polishing of optical-quality glass.

Some of the rare-earth metals (and their atomic weights) that are commonly used in electronics include lanthanum (57), cerium (58), neodymium (60), samarium (62), europium (63), terbium (65), and dysprosium (66).

In the 1990s, mining companies in China began focusing on rare earths; mines in other countries could not compete with low-cost Chinese mining and processing.

The demand for rare earths is expected to increase over the next several years, driven by an increase in the use and production of items that are manufactured using rare earths. For example, in 1998, cellular telephones, which have batteries that require rare earth elements, were used by just 5.3% of the global population, according to International Telecommunications Industry data. By 2017, the penetration rate of cellphones worldwide had reached 103.4% (exceeding 100% due to ownership of multiple devices).

Other products, such as electric vehicles and wind turbines, were just in the prototype phase two decades ago, but have since seen significant commercial deployment, with positive demand forecast over the next several decades. As a result, demand for rare earth elements is likely to grow over time; combined with a relatively limited base of suppliers, rising demand could drive up the cost of rare earths for manufacturers, both in the U.S. and around the globe.

As recently as 30 years ago, rare earths were mined and processed in various countries around the world, including the U.S. However, in the early 1990s, Chinese-based mining companies began developing rare-earths operations, and other mines around the world simply could not compete, due to the lower cost of operations at the Chinese mines and processing facilities. As a result, Chinese-based companies wound up controlling more than 90% of the market by the late 1990s, and prices remained relatively steady.

However, in 2010, China cut its export quotas for rare earth exports, and rare earth prices skyrocketed. Furthermore, a territorial dispute with Japan led China to halt exports to that country for two months, proving its control over rare earths could also be a weapon in any sort of international dispute. Speculators hoarded rare earth minerals, sending prices soaring. Seeing that the Chinese government was actively using its control over the rare earths market to get what it wanted, new rare earth production facilities were started in the U.S., Australia, Russia, Thailand, Malaysia, and other countries.

A potential trade war between China and the U.S., a net importer of rare earths, combined with an expected rise in demand for rare earths over the next few years, could contribute to rising rare earth prices.

When prices of rare earths spiked in the past, manufacturers were able to get their engineers to reduce the requirements for rare earths in some products, such as reducing or eliminating the use of europium and terbium in fluorescent lighting products, says Pierre Neatby, vice president of sales and marketing with Avalon Advanced Materials, Inc., a Canadian mineral development company with three mining projects expected to enter commercialization, including rare earth elements tantalum, niobium and zirconium.

A new red phosphor that uses Manganese4+ (Mn4+) activated fluoride compounds was developed to replace rare earth materials in lighting. However, Neatby says, some products simply require rare earths in order to provide the level of performance demanded by customers.

Indeed, suitable substitutes for neodymium magnets, which are valued because they are extremely powerful and lightweight, have yet to be found, Neatby says. "So, the neodymium magnet is still the most powerful magnet in the world, and for [electric] car applications, you do want the smallest, lightest, motor, because the heavier the car, the bigger the engine has to be in order to move it forward," Neatby says. "Whether it's an F-35 [fighter jet], a big submarine, or electric car, rare earth magnets are going to be used."

Production of military equipment, such as the aforementioned F-35, is at risk of being impacted by the escalating U.S.–China trade war, as are other key pieces of military equipment, such as night-vision goggles, precision-guided weapons, communications gear, GPS systems, batteries, and other defense electronics, each of which requiring rare earth metals as key ingredients.

A provision in the National Defense Authorization Act signed into law in August 2018 bars the U.S. Department of Defense (DoD) from buying permanent rare-earth magnets made in China after December 2018. As there is no domestic source of the materials used to make those magnets large enough to support current demand, the measure will likely force the DoD to purchase rare earths from Japanese producers of rare-earth elements, the main source for rare earths outside China.

The U.S. Geological Survey estimated U.S. manufacturers consumed 11,000 tons of rare earth elements in 2017. Although the Trump administration backed down from imposing steep tariffs on rare-earth elements from China after appeals from U.S industrial consumers of the elements, China has not reciprocated, and U.S. rare earth exports have been hit with a 10% duty now, which could rise to 25% later this year. This tariff is expected to negatively impact the sole U.S. rare earth mine in operation, located at Mountain Pass, CA, and the ability of the U.S. manufacturers to reestablish a self-sufficient rare earths manufacturing industry.

The Mountain Pass mine was acquired last year by U.S.-owned consortium MP Mine Operations LLC, after being shuttered since 2015 by its previous owners, MolyCorp. Inc., when that firm went bankrupt. Mountain Pass operations were restarted in 2018, with the mine's rare earth compounds now being shipped to China for separation into individual rare-earth oxides, though steep tariffs are likely to negatively impact the company's ability to compete against Chinese producers.

"The Chinese are still the kings of downstream production of materials that use rare earths; Molycorp has gone bankrupt, and the only outside significant producer of rare earths is Lynas (Corporation of Australia), but they're producing essentially light rare earths," Neatby says. "All of the heavy rare earths still come from China, and you have a situation where electric vehicles in the future are going to use electric motors that use rare earths, and demand is starting to pick up."

Due to the threat of the supply of rare earths being cut off, many companies have stockpiled larger supplies of rare earths. Despite the forces impacting the supply side of the market, demand is likely to remain strong.

"In general, you've got a rare earth market that is quite solid," Neatby says. "All of the applications that were probably not economic have gone away, and so now [the market] is efficient, and focused on neodymium and presidium, and maybe a bit of dysprosium for high-temperature magnets. And traditional applications for lanthanum, cerium, for lanthanum for cracking catalysts for oil and gas, continue to be used. Those are bigger volumes, but [account for] less value."

The most prudent strategy to ensuring the steady supply of rare earths to manufacturers around the world is to increase mining and refining outside of China. The only current major rare earths producer outside of China is Lynas, which operates the Mount Weld mine in Western Australia, and produces more than 5,000 metric tons of neodymium and praseodymium (NdPr) per year, with most of its output committed to Japanese buyers. Other projects in Australia, Russia, and Brazil are set to enter production over the next several years.

Moreover, there was a large discovery last year of hundreds of years' worth of rare-earth materials underneath Japanese waters, essentially enough to supply to the world on a "semi-infinite basis," according to a study published in Nature Publishing Group's *Scientific Reports*.

However, the major challenge involves efficiently and cost-effectively separating these rare earths, and a consortium of Japanese government-backed entities, companies, and researchers plans to conduct a feasibility test within the next five years.

Still it is likely that, regardless of the outcome of any trade negotiations to reduce or eliminate tariffs, or the development of a new way to extract rare earths from new deposits, U.S. manufacturers' best hope for securing rare earths will be the redevelopment of a domestic rare earths industry.

A source close to MP Mine Operations said that the company's operating plan is to create processing capacity to allow a full, end-to-end mining, extraction, and processing capability in the U.S. within 18 months, which may help to alleviate the pressure on the market. According to a source close to the company, MP Mine Operations is trying to create an American supply chain and is hoping the U.S. government will pressure China to reduce or eliminate import tariffs that affect the company.

James Litinsky, chief executive officer of JHL Capital Group LLC, the majority owner of the Mountain Pass consortium, told Bloomberg News last September that Mountain Pass' "self-sufficiency will serve as a foundation for an American-based rare earths industry." C

Further Reading

Rare Earth Technology Alliance,
What Are Rare Earths?,
http://www.rareearthtechalliance.com/What-are-Rare-Earths

Coyne, K.
Moving Past Neodymium: Scientists Explore Alternative to Expensive Rare Earth Element, *R&D Magazine*, September 27, 2017, http://bit.ly/2OUwHkK

Video: Super Elements BBC Documentary on Rare Earths (2017): https://www.youtube.com/watch?v=GOBKa4vxAOE

Keith Kirkpatrick is principal of 4K Research & Consulting, LLC, based in Lynbrook, NY, USA.

SHAPE THE FUTURE OF COMPUTING.
JOIN ACM TODAY.

www.acm.org/join/CAPP

SELECT ONE MEMBERSHIP OPTION

ACM PROFESSIONAL MEMBERSHIP:

- ❑ Professional Membership: $99 USD
- ❑ Professional Membership plus
  ACM Digital Library: $198 USD
  ($99 dues + $99 DL)

ACM STUDENT MEMBERSHIP:

- ❑ Student Membership: $19 USD
- ❑ Student Membership plus ACM Digital Library: $42 USD
- ❑ Student Membership plus Print *CACM* Magazine: $42 USD
- ❑ Student Membership with ACM Digital Library plus
  Print *CACM* Magazine: $62 USD

❑ **Join ACM-W:** ACM-W supports, celebrates, and advocates internationally for the full engagement of women in computing. Membership in ACM-W is open to all ACM members and is free of charge.

PAYMENT INFORMATION

Name

Mailing Address

City/State/Province

ZIP/Postal Code/Country

❑ Please do not release my postal address to third parties

Email Address

❑ Yes, please send me ACM Announcements via email
❑ No, please do not send me ACM Announcements via email

❑ AMEX ❑ VISA/MasterCard ❑ Check/money order

Credit Card #

Exp. Date

Signature

Purposes of ACM

ACM is dedicated to:

1) Advancing the art, science, engineering, and application of information technology

2) Fostering the open interchange of information to serve both professionals and the public

3) Promoting the highest professional and ethics standards

By joining ACM, I agree to abide by ACM's Code of Ethics (www.acm.org/code-of-ethics) and ACM's Policy Against Harassment (www.acm.org/about-acm/policy-against-harassment).

I acknowledge ACM's Policy Against Harassment and agree that behavior such as the following will constitute grounds for actions against me:

- Abusive action directed at an individual, such as threats, intimidation, or bullying
- Racism, homophobia, or other behavior that discriminates against a group or class of people
- Sexual harassment of any kind, such as unwelcome sexual advances or words/actions of a sexual nature

BE CREATIVE. STAY CONNECTED. KEEP INVENTING.

acm Association for Computing Machinery

ACM General Post Office
P.O. Box 30777
New York, NY 10087-0777

1-800-342-6626 (US & Canada)
1-212-626-0500 (Global)
Hours: 8:30AM - 4:30PM (US EST)

Fax: 212-944-1318
acmhelp@acm.org
acm.org/join/CAPP

DOI:10.1145/3306610

Pamela Samuelson

Legally Speaking

Questioning a New Intellectual Property Right for Press Publishers

Considering the implications of the "link tax" provision of the proposed EU Directive for the Digital Single Market for traditional press publishers.

SHOULD EUROPEAN PRESS publishers be granted a new intellectual property (IP) right over online uses of their journalistic contents? These publishers have long had both copyright and sui generis database IP protections for these contents. Yet the European Commission, Council, and Parliament have been convinced that only by granting the new IP right will sustainable quality journalism continue to be produced in the EU. Despite some strong opposition, this proposal seems likely to be adopted and made a mandatory part of EU law.

Sometimes known by its critics as the "link tax" provision, Article 11 of the proposed Directive for the Digital Single Market (DSM) would grant press publishers a new set of exclusive rights to control the reproduction and making available of online journalistic contents by information society service providers. Under the proposed compromise text made public in November 2018, these rights would last for one year.

Critics of Article 11 have tried to blunt somewhat the scope of this new right. For instance, the Parliament's version of Article 11 would provide that the right "shall not extend to mere hyperlinks which are accompanied by individual words." But the Council and the Commission have not exactly agreed to this change or to the Parliament's proposed exception for "legitimate private and noncommercial uses" by individual users; negotiations to finalize the text of this Directive are ongoing and likely to be concluded in 2019.

Arguments for the Press Publisher Right

It is no secret that these are trying times for press publishers. Paid subscriptions have generally declined, readership has eroded, and advertising revenues that long supported print journalism have shrunk. The transition to digital publishing has been challenging and required experimentation with new business models. Press publishers are fearful these business models will not suffice to sustain their industry.

The moral argument said to support the new press publishers' right arises from a sense of unfairness that technology companies (think Google) and online news aggregator services (for example, Meltwater) are making money, either from advertising or from subscriptions, by providing members of the public with free access to their news, through links and snippets, without compensating the publishers who provided that news.

A secondary argument has focused on difficulties that press publishers have sometimes encountered in proving copyright ownership in articles written by freelancers when suing search engines or news aggregators.

Some momentum for the press publisher right has built up in the last few

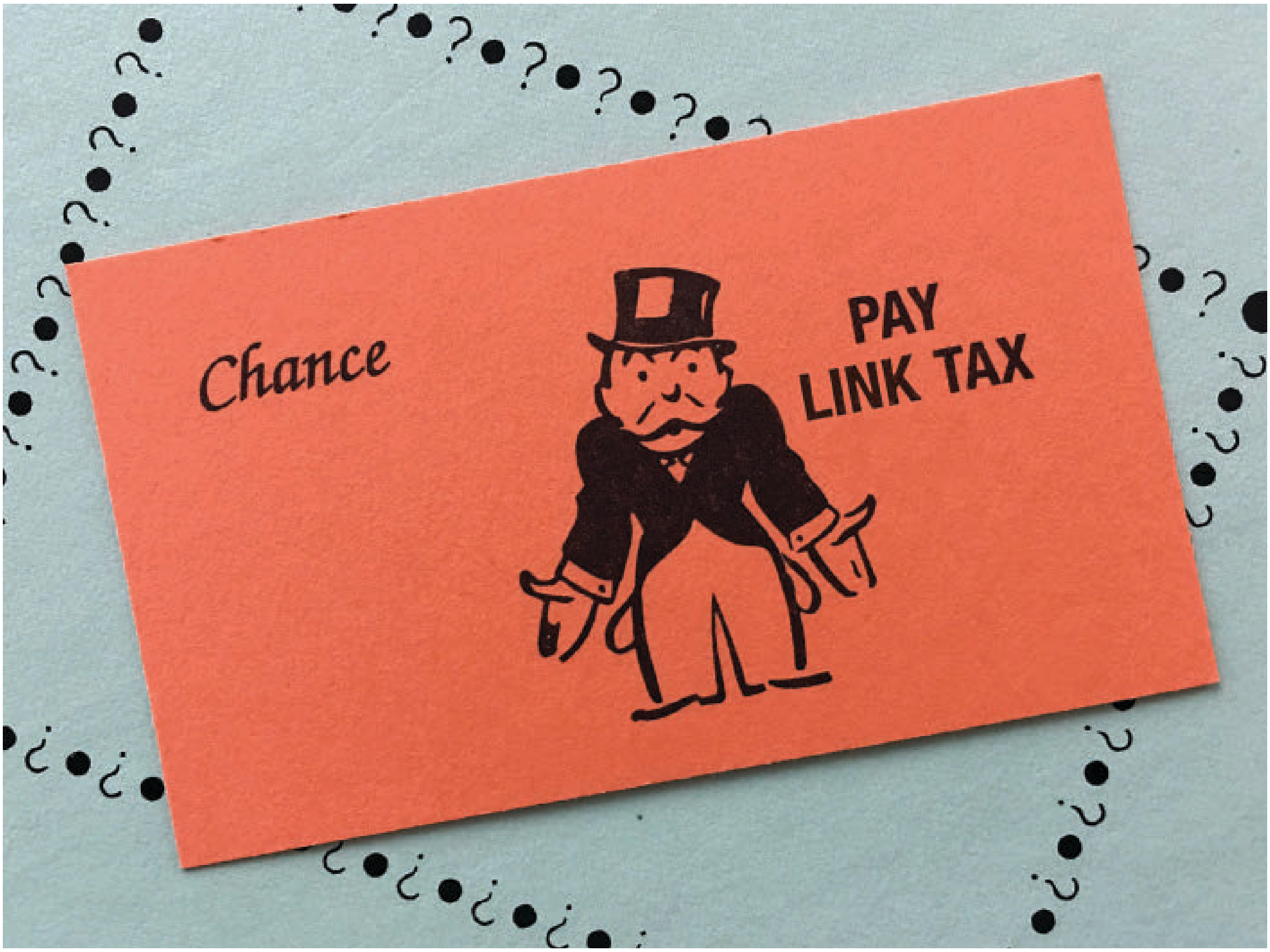


IMAGE BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

years, taking advantage of a general sense of hostility in the EU toward major technology U.S. companies. Google, Apple, Facebook, and Amazon (aka GAFA) are chief among them.

Complaints are legion that these firms have abused their dominant positions, been responsible for fake news and privacy breaches, and/or shown indifference toward other firms' IP rights. EU policymakers have been persuaded that there is a "value gap" that digital technology companies should fill by licensing content from EU press publishers, among others.

Lessons from Germany and Spain

A few years before the DSM Directive was proposed, press publishers persuaded the German and Spanish legislatures (in 2013 and 2014 respectively) to pass laws granting them rights similar to those that Article 11 would mandate for all EU member states. These laws have met with much less success than their proponents had hoped.

After the German law passed, press publishers authorized VG Media, their collecting society, to establish 6% of gross revenues as the license fee it should collect from technology firms for rights to make online uses of the publishers' journalistic contents. According to a report commissioned by a European Parliament committee (whose lead author is the well-known copyright scholar Lionel Bently), Google refused to pay such a fee and eventually got some free licenses.[1]

These licenses notwithstanding, VG Media sued Google for violating this right. A German court stayed the proceedings so that the Court of Justice of the EU (CJEU) could address a question about the validity of the law. As of 2017, the society had issued only five licenses and collected a total of 714,000 euros.

Google also refused to pay license fees to contents of Spanish press publishers. Instead it shut down its Spanish Google News service. Seemingly as a result, estimated traffic to Spanish news sites declined by somewhere between 6% and 30%. Search engines and news aggregators, it turned out, had driven traffic to Spanish news sites. Very few licenses have issued under the Spanish law.

Why Many Oppose the Proposed New IP Right

In April 2018, a group of 169 IP academics sent a statement to the EU Parliament strongly opposing Article 11.[4] There was, it said, "no indication whatsoever that the proposed right will produce the positive results it is supposed to." Moreover, "considering the current high levels of market concentration on online advertising markets and in media, a publishers' right may well backfire: further strengthening the power of media conglomerates and of global platforms to the detriment of smaller players." (By July 2018, another 69 IP academics joined this letter.)

Article 11 would, the Statement concluded, impede the free flow of news and other information vital to a democratic society, harm journalists who often rely on search engines and aggregators, and create uncertainty about its coverage and scope.

It was also unclear how the new publisher right would interact with existing copyright laws, which typically allow for fair quotations, and database rights, which allow extractions of insubstantial parts of databases.

The economic case for Article 11 was, moreover, weak. The press publisher right would increase transaction costs considerably, as well as exacerbating existing power asymmetries in media markets. The Statement pointed to the ineffectiveness of the German and Spanish press publisher regimes as additional reasons not to create such an EU-wide right.

The Max Planck Institute's Center for Innovation and Competition also published a Position Statement opposed to Article 11.[3] The European Copyright Society's response to the European Commission's public consultation on the role of publishers in the copyright value chain raised significant questions about the proposed press publishers' right.[2]

The Bently et al. report noted that online journalists perceive the new right as a threat to the nature of news communication in the modern era: "Paying for links is as absurd as paying for citations in the academy would be." That report cast doubt on the wisdom of adopting a provision such as Article 11.

Will Compromise Provisions Overcome Opposition?

To respond to concerns expressed by various critics, the European Parliament in September 2018 approved several amendments to Article 11. For instance, it proposed creating an exception for individual users to make "legitimate private and noncommercial uses" of press contents.

The compromise text made public in November 2018 contains a similar, although differently worded, provision. It states that the press publisher rights "shall not apply to uses of press publications carried out by individual users when they do not act as information society service providers." This is, however, still quite vague. Would it, for instance, exempt a person or nonprofit organization that regularly blogs with links to EU press publisher sites?

It remains unclear whether hyperlinking to press publisher contents will generally be lawful.

The Parliament-approved version also provided that "mere hyperlinking accompanied by individual words" would not trigger liability. However, the latest compromise text has retained the Commission's original version of Article 11, which would extend liability to hyperlinking if it constituted a communication to the public.

Because the communication right in respect of hyperlinking is an ever-evolving concept under some very confusing CJEU decisions, it remains unclear whether hyperlinking to press publisher contents will generally be lawful.

Yet, the November 2018 compromise text would qualify the press publisher right by providing that "uses of insubstantial parts of a press publication" should not give rise to liability. Moreover, member states could determine what parts of press publications are "insubstantial" by "taking into account whether these parts are the expression of the intellectual creation of their authors, or whether these parts are individual words, or very short excerpts." This qualification is better than nothing, but notice how vague is the concept of "insubstantial" and what if member states differ on how many words are too many?

Another qualification proposed in a recital to the compromise text indicates the rights should not extend to "mere facts" reported in the press publications. Again, this is better than no such limitation, but it begs the question of what "mere facts" includes and does not include.

The Parliament's version of Article 11 would have cut the duration of the proposed press publisher right to a five-year term in contrast to the Commission's original proposal of 20 years from publication. But when it comes to news, even five years seems unduly long. The November 2018 compromise text would follow the German law in granting press publishers these rights for only one year.

Finally, the Parliament-approved version of Article 11 proposed requiring press publishers to provide authors with a "proportionate" share of whatever revenues the publishers collect from licensees of the new right. This might well reduce (perhaps by half) the benefits to publishers from creation of this new IP right. Or it may instead lead to much higher license fees to fund the author-sharing. The November 2018 compromise text retains this proposal.

As well-meaning as the author-sharing proposal may be, it underestimates how substantial will be the costs necessary to obtain sufficient information to determine which authors are entitled to get what part of each press publisher's revenues.

Conclusion

A key assumption underlying the proposed DSM Directive, including Article 11, is that strengthening European IP rights will lead to much greater licensing revenues flowing to European rights holders from (mostly) American technology companies. (See my November 2018 column discussing the even more controversial Article 13 of this Directive.)

European press publishers have lobbied heavily for this new right and seem on the verge of getting a significant boost in leverage this right will give them to negotiate for new revenue streams from search engines and news aggregators. The German and Spanish experiences thus far cast doubt on the prospects for significant successes. Whether an EU-wide right will achieve better results remains to be seen. Maybe Google and Facebook will pay up, but maybe not.

There is, of course, some irony in the EU's prospective adoption of a Directive aimed at promoting a "digital single market" given that no one licensing entity exists from which technology firms can get an EU-wide license. Each member state will have its own implementation of the DSM directive. Prospective licensees will have to negotiate with every member states' preferred collecting society to clear all the rights necessary to make digital uses of European journalistic contents.

Even if Google and Facebook decide to take licenses from European press publishers and can afford to negotiate all of the necessary licenses, isn't there a significant risk these licenses will further entrench them as dominant players in global information markets? The new press publisher right would seem to impose significant transaction costs as well as establish expensive licensing fees for some individual bloggers, innovative startups, and small enterprises that may want to link to journalistic contents from European sites.

While there is very little chance at this point that Article 11 will be deleted from the DSM Directive, some further compromises may be negotiated by those responsible for finalizing its text so that freedom of information and expression are not unduly repressed by adoption of this unfortunately ambiguous new IP right.

A closed-door "trilogue" is under way among representatives of the European Commission, the Council, and the Parliament, each of which has supported a different version of Article 11. The November 2018 compromise text will likely not be the last word. Other nations should, however, be wary of following the EU's lead on this particular initiative.

References

1. Bently, L. et al. Strengthening the Position of Press Publishers and Authors and Performers in the Copyright Directive: A Study Commissioned by the European Parliament (2017); https://bit.ly/2wL9ZhQ
2. Kretschmer, M. et al. The European Commission's Public Consultation on the Role of Publishers in the Copyright Value Chain: A Response by the European Copyright Society, European Intellectual Property Review (E.I.P.R.) 38, 10 (Oct. 2016), 591–595; https://bit.ly/2LuOHgH.
3. Max Planck Institute for Innovation and Competition. Position Statement on Proposed Modernisation of European Copyright Rules, Part E Protection of Press Publications Concerning Digital Uses; https://bit.ly/2EHa6mb
4. Ricolfi, M., Xalabarder, R., and van Eechoud, M. Academics Against Press Publishers' Right, Statement from 169 EU Academics, 2018; https://bit.ly/2r3a1QD.

Pamela Samuelson (pam@law.berkeley.edu) is the Richard M. Sherman Distinguished Professor of Law and Information at the University of California, Berkeley, and a member of the ACM Council.

Calendar of Events

March 10–14
CHIIR '19: Conference on Human Information Interaction and Retrieval,
Glasgow, U.K.,
Sponsored: ACM/SIG,
Contact: Martin Halvey,
Email: martin.halvey@gmail.com

March 17–20
IUI '19: 24th International Conference on Intelligent User Interfaces,
Los Angeles, CA,
Co-Sponsored: ACM/SIG,
Contact: Wai-Tat Fu,
Email: wfu@illinois.edu

March 17–20
TEI '19: 13th International Conference on Tangible, Embedded, and Embodied Interaction,
Tempe, AZ,
Sponsored: ACM/SIG,
Contact: Stacey Kuznetsov,
Email: kstace@asu.edu

March 21–22
TAU '19: ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems,
Monterey, CA,
Sponsored: ACM/SIG,
Contact: Song Chen,
Email: songchen@pacbell.net

March 25–27
CODASPY '19: 9th ACM Conference on Data and Application Security and Privacy,
Richardson, TX,
Sponsored: ACM/SIG,
Contact: M.B. Thuraisingham,
Email: bhavani.thuraisingham@utdallas.edu

March 25–28
EuroSys '19: 14th EuroSys Conference 2019,
Dresden, Germany,
Sponsored: ACM/SIG,
Contact: Christof Fetzer,
Email: cf2@inf.tu-dresden.de

April

April 3–4
SOSR '19: Symposium on SDN Research,
San Jose, CA,
Sponsored: ACM/SIG,
Contact: Eric Rozner,
Email: erozner@gmail.com

DOI:10.1145/3306615

Ofir Turel

▸ Marshall Van Alstyne, Column Editor

Economic and Business Dimensions

Potential 'Dark Sides' of Leisure Technology Use in Youth

Time for balanced reflections on technology.

COMPUTING TECHNOLOGY HAS produced many societal benefits. Nevertheless, it often serves as a double-edged sword and promotes negative consequences, such as distraction, addiction, time waste, and reduced well-being.[10] This is perhaps not surprising given that "When you invent the ship, you also invent the shipwreck ... Every technology carries its own negativity, which is invented at the same time as technical progress."[11] Indeed, many computing technologies follow this pattern, exhibiting a duality of "bright" and "dark" effects on people, firms, and societies.[3,4] The problem is that the understanding of downsides of technology sometimes lags our understanding of upsides. We, especially technology enthusiasts, are often enchanted by the abundant positive things new technologies can do, and this dilutes our ability to develop reliable judgments regarding the harms new technologies can cause.[7]

IMAGE BY MIKE SHOTS

While studies of both positive[5] and negative[9] technology effects on children and youth exist, trends in technology use among youth and their possible adverse associations are less well explored. It is important to examine and discuss such trends given the lack of regulation of technology use and the limited awareness to possible technology use harms. By contrast with other harmful materials and behaviors (for example, using illicit substances, consuming junk food, not wearing seatbelts), the use of computing technologies is largely unregulated and many parents and children may not be aware of the extent of harm that may be associated with excessive use of technology. Hence, analyzing such trends can serve as a springboard for initiating a healthy discussion in our discipline regarding possible "dark side" effects of computing technologies on youth, and more broadly speaking, developing a more balanced discussion regarding the effects of technologies on societies. Increasing awareness to such issues and sparking this discussion are needed steps before we mobilize resources and develop and test solutions for possible largely unexpected negative effects of technology use on youth.

Here, I seek to shed light on technology use trends in youth and examine their parallels with a range of adverse outcomes in the school, social, well-being, and health domains. To achieve this objective, I analyze a large dataset (n=152,172) of survey responses by youth, approximately 13–16 years old, across the U.S. This data is drawn from an annual (2012–2016) survey administered to hundreds of schools.[2]

Results

Figures 1–4 portray, correspondingly, trends in: time (hours/day) spent on leisure vs. for school computing and work; healthy lifestyle activities; social activities; and well-being and self-worth. Error bars represent 95% confidence intervals.

Figure 1 demonstrates an increase in the use of computing technologies, both for leisure and for school purposes. However, the average increase in technology use for leisure (30 minutes/day) is twice as much as the average increase in the use of technology for school assignments (15 minutes/day). Given the zero-sum-game of a student's after-school time, one possibility is the use of technologies for leisure purposes is alluring and consequently cannibalizes from schoolwork time (average reduction of 11.4 minutes/day between 2012 and 2016). Another possibility is the changes in the use of technology for schoolwork increases efficiency in homework tasks; but the nature of such potential efficiencies (for example, increased ease of finding explanations vs. increased ease of finding an online solution to copy) is unclear.

Figure 2 shows the changes in technology use patterns described in Figure 1 parallel a decline in the frequency of important healthy lifestyle activities, including eating breakfast, exercising, and getting sufficient sleep. Again, this might be explained via the zero-sum-game argument; the use of alluring technologies might have cannibalized from healthy lifestyle activities. For instance, video gaming can consume people's time and prevent physical activity; it can also reduce sleep via the blue light emitted from screens.[8] Moreover, in many cases young video gamers deceive their parents and play late at night or early in the morning,[1] which can explain reduction in breakfast frequency.

Figure 3 shows a decline in face-to-face social activities in youth that parallels the increase in the use of leisure technologies. Circa 2016, youth attended social functions, met friends, and went on dates less frequently compared to 2012 youth. Technology can

Figure 1. Trends in time spent on leisure computing vs. school work in youth.

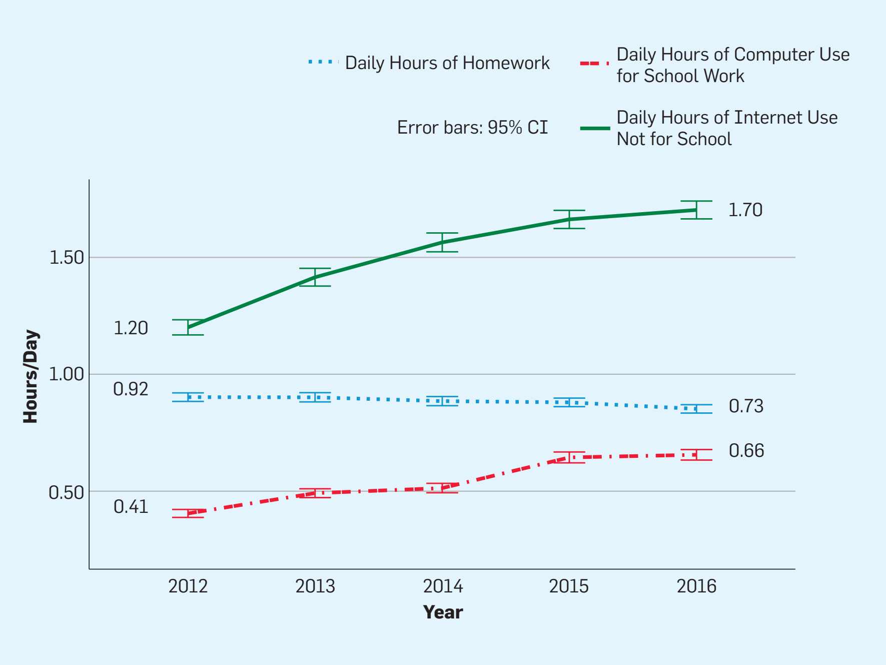
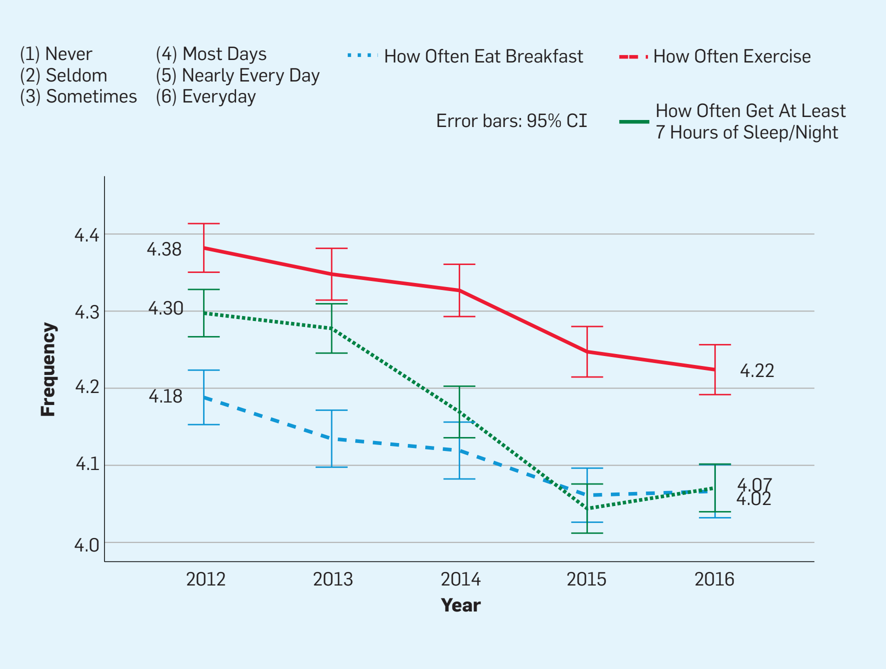


Figure 2. Trends in healthy lifestyle activities in youth.



serve as one explanation for this decline as it can isolate youth, build online socialization habits, and reduce youth's motivation and ability to interact face-to-face.[10,12]

Lastly, Figure 4 demonstrates a general decline in well-being and self-worth perceptions that parallels the increase in leisure technology use. This can be explained via the increase in use of social media, where everyone else's life seems perfect. Social media users are exposed to a larger comparison set. If a child is comparing him- or herself to the top people of an ever-expanding set, then it is conceivable that he or she might experience a growing inferiority complex.

The parallels between these groups of trends can of course be a coincidence. However, it is also possible that, as per the many studies indicating possible negative effects of technologies on adults and young adults,[4,7,10] the sometimes excessive use of leisure technologies in youth can adversely affect school, social, health and well-being facets. While I could not calculate all correlation given the nature of the dataset, existing correlations provide some support for these claims. The hours/day of use of the Internet for leisure purposes was significantly negatively correlated with the frequency of meeting friends informally, face-to-face (r = -0.025), and with attending social functions (r=-0.010). Thus, it can be viewed as a correlate of reduction in face-to-face social activities. The use of leisure technologies was also positively correlated with the use of technology for school work (r=0.198). Thus, it is possible that encouraging youth to use technology for school work backfires, as the mere presence of a computer may allure them to spend more leisure time on the Internet. Note the relatively small correlations imply the use of technology for leisure purposes may not be the only or prime cause for adverse outcomes in the social domain, but it may be viewed as a potentially contributing factor for such issues.

Figure 3. Trends in social activities in youth.

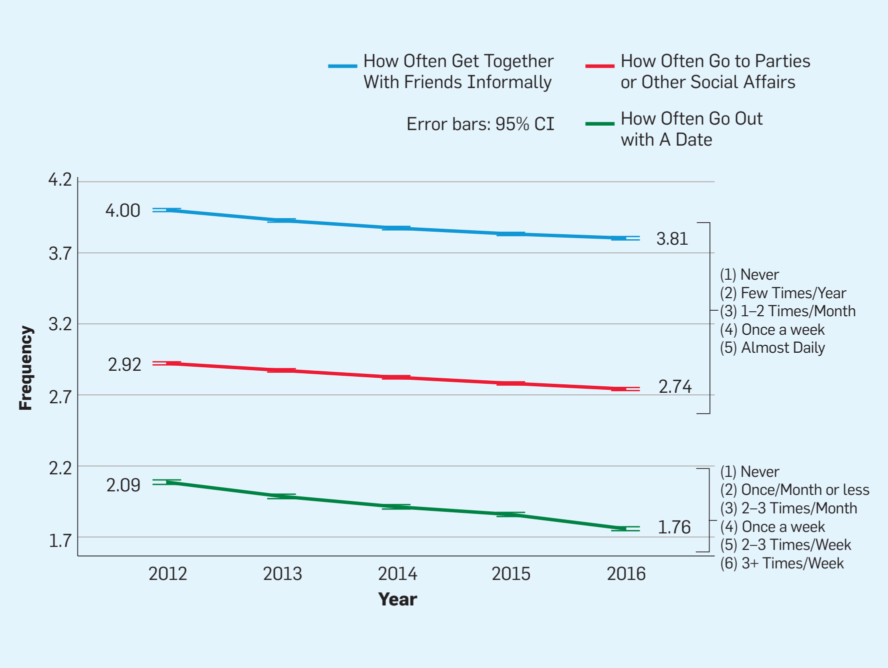
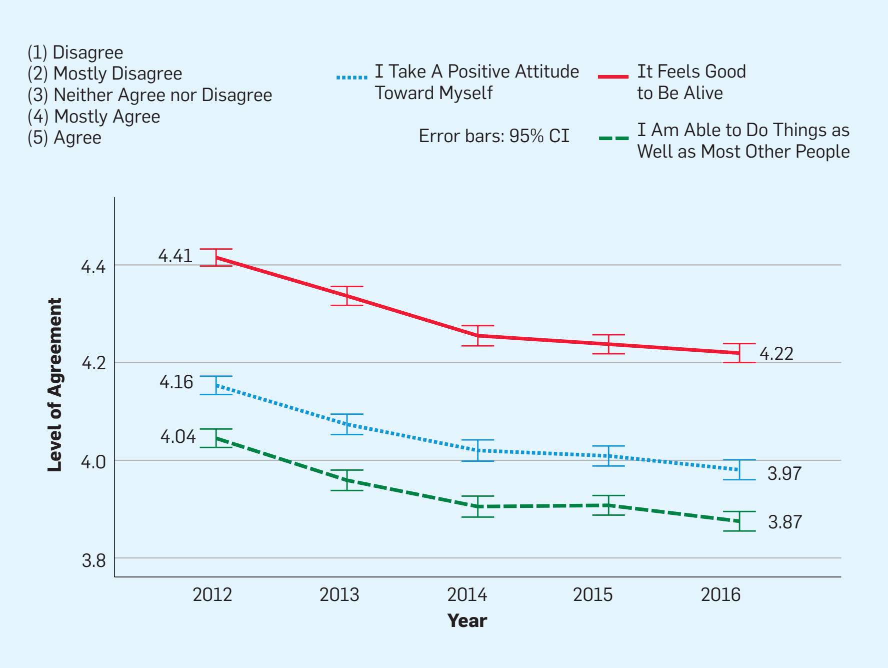


Figure 4. Trends in well-being and self-worth in youth.



Time for Balanced Reflections on Technology

For many years we have emphasized the positive aspects of computing technologies because we believed in their contribution to humanity. Nevertheless, there is a growing body of evidence in support of a technology duality view. That is to say, we have started realizing and quantifying the notion that many of the technologies we develop can also be harmful, especially when used excessively. While adults can typically understand and deal with such issues, for example, through self-regulation of leisure technology use during work hours, youth often cannot do so as effectively. This difference stems from the idea that their brains are still developing, and the parts of the brain that drive rewarding behaviors develop faster than their brain regions that are involved in self-control.[6] It is hence our responsibility to better inform them, their families, and their educators regarding possible risks that may be associated with improper and excessive use of leisure computing technologies. We

Many of the technologies we develop can also be harmful, especially when used excessively.

should also consider whether the leisure computing technologies we develop should allow users to: easily self-track own activity; inform young users and parents when dangerous levels of activity are reached; and restrict/block activities by request.

It is informative to reflect on parallels between our industry (the tech-sector) and other industries that revealed "dark sides" after a period of focusing almost solely on positive aspects of their products. Two industries that come to mind are the tobacco and food industries. The tobacco industry sold its products and emphasized their positive effects (for example, increased concentration) while hiding its negative effects. Court rulings have forced it to pay restitution, and regulations have forced it to restrict the use of tobacco products to adults, and to advertise the risks associated with its use. Consequently, there has been a constant decline in tobacco consumption in Western countries.

Perhaps this is an extreme parallel, because one can argue that people can live without tobacco, but technology is essential to functioning in modern society. If so, consider the food industry parallel. On an evolutionary timescale, food was scarce so people developed innate preference for fatty and sugary foods. Modern ability to satiate these needs has improved and companies have created many such foods to the point where unhealthy food is abundant and obesity became an epidemic. Governments regulate food by enforcing food labeling as a means to inform consumers. Simultaneously, awareness regarding proper nutrition has increased. The responsibility for possible overconsumption, in this case, lies with parents, educators, and children. Adapting this view, we could argue for increasing awareness regarding leisure technology use risks and at the same time ask the developers of such technologies to either voluntarily or through government regulation provide people with the means to track use and to have more usage control and self-monitoring. This is, for example, exactly what Apple has done with iOS 12. Still, we cannot solely count on tech providers; the responsibility to inform our children regarding such risks and to teach them to live responsibly with technology is likely still ours. As a discipline, we certainly must start developing a balanced discussion that acknowledges both possible positive and negative effects of technology on children and young adults.

References

1. Bruner, O. and Bruner, K. *Playstation Nation: Protect Your Child from Video Game Addiction*. Hachette Book Group, NY, 2006.
2. Johnston, L.D. et al. *Monitoring the Future: A Continuing Study of American Youth (8th- and 10th-Grade Surveys)*, 2015. Research, I.f.S. Ed., Inter-university Consortium for Political and Social Research (ICPSR), Ann Arbor, MI, 2016.
3. Malhotra, A. and Van Alstyne, M. The dark side of the sharing economy … and how to lighten it. *Commun. ACM 57*, 11 (Nov. 2014), 24–27.
4. Tarafdar, M. et al. The dark side of information technology. *MIT Sloan Management Review 56*, 2 (Winter 2015), 600–623.
5. Turel, O. and Bechara, A. Little video-gaming in adolescents can be protective, but too much is associated with increased substance use. *Substance Use and Misuse*. (Jan. 2019); DOI: 10.1080/10826084.2018.1496455
6. Turel, O. et al. An examination of neural systems sub-serving Facebook "addiction." *Psychological Reports 115*, 3 (Mar. 2014), 675–695.
7. Turel, O. and Qahri-Saremi, H. Problematic use of social networking sites: Antecedents and consequence from a dual system theory perspective. *Journal of Management Information Systems 33*, 4 (Apr. 2016), 1087–1116.
8. Turel, O., Romashkin, A., and Morrison, K.M. Health outcomes of information system use lifestyles among adolescents: Videogame addiction, sleep curtailment and cardio-metabolic deficiencies. *PLoS One 11*, 5 (May 2016), e0154764.
9. Turel, O., Romashkin, A., and Morrison, K.M. A model linking video gaming, sleep quality, sweet drinks consumption and obesity among children and youth. *Clinical Obesity 7*, 4 (Apr. 2017), 191–198.
10. Turel, O. and Serenko, A. The benefits and dangers of enjoyment with social networking websites. *European Journal of Information Systems 21*, 5 (May 2012), 512–528.
11. Virilio, P. and Petit, P. *Politics of the Very Worst*. Semiotext(e), NY, 1999.
12. Xu, Z.C., Turel, O. and Yuan, Y.F. Online game addiction among adolescents: Motivation and prevention factors. *European Journal of Information Systems 21*, 3 (Mar. 2012), 321–340.

Ofir Turel (oturel@fullerton.edu) is a professor of Information Systems and Decision Sciences at California State University, Fullerton, and a scholar in residence at the decision neuroscience lab at the University of Southern California; http://oturel1.wixsite.com/ofirturel

DOI:10.1145/3306614

Peter J. Denning

The Profession of IT
An Interview with William Hugh Murray

A discussion of the rapidly evolving realm of practical cyber security.

William Hugh (Bill) Murray is a management consultant and trainer in Information Assurance specializing in policy, governance, and applications. He has more than 60 years experience in information technology and more than 50 years in security. During more than 25 years with IBM his management responsibilities included development of access control programs, advising IBM customers on security, and the articulation of the IBM security product plan. He is the author of the IBM publication *Information System Security Controls and Procedures*. He has been recognized as a founder of the systems audit field and by *Information Security Magazine* as a Pioneer in Computer Security. He has served as adjunct faculty at the Naval Postgraduate School and Idaho State University. In 1999, he was elected a Distinguished Fellow of the Information System Security Association. In 2007, he received the Harold F. Tipton Award in recognition of his lifetime achievement and contribution. In 2016, he was inducted into the National Cyber Security Hall of Fame. In 2018, he was elected a Fellow of (ISC)^2—see https://www.isc2.org/).

Bill Murray has been responding for years to security threats with nonconventional thinking. When he sees a security breakdown, he asks what is the current practice that allows the breakdown to happen, and what new practice would stop it? Most of our security

IMAGE COURTESY OF (ISC)2 BLOG

vulnerabilities arise from poor practice, not from inadequate technology.

Many people today are concerned about cybersecurity and want to know how to protect themselves from malware, identity thieves, invading hackers, botnets, phishers, and more. I talked to Bill about what practices we have to deal with these issues, and where we need to look for new practices.

Q: Weak passwords have been the bane of security experts for years. Early studies of time-sharing systems showed that in a community of 100 users, two or three are likely to use their own names as passwords. A hacker can break in easily if passwords are so easy to guess. You declared that the root cause of this is the reusability of passwords. You proposed that we use technologies where a password can be used only once. How does this work and why is it now feasible?

A: This is not simply about "weak passwords" but all passwords. It is time to abandon passwords for all but trivial applications. Passwords are fundamentally vulnerable to fraudulent reuse. They put the user at risk of fraudulent use of identity, capabilities, and privileges and the system or application at risk of compromise and contamination by illicit users. Strong passwords protect against brute force attacks but these are not the attacks that we are seeing.

We need "strong authentication," defined as at least two kinds of evidence of identity, one resistant to brute force attacks and the other resistant to replay, that is, includes a one-time value. All strong authentication is "multi-factor" but not all multi-factor is strong. Strong authentication protects us against both brute force attacks and the fraudulent reuse of compromised credentials, for example from so called "phishing" attacks, the attacks that we are actually seeing.

Steve Jobs and the ubiquitous mobile computer have lowered the cost and improved the convenience of strong authentication enough to overcome all arguments against it.

Q: The Internet is seen as a flat network where any node can communicate with

any other. One of the fundamental ideas baked into the Internet protocols is anonymity. This presents immense problems for local networks that want to be secure because they cannot easily validate whether requested connections are from authorized members. What technologies are available to define secure subnets, abandoning the idea of flatness and anonymity?

A: The Internet is flat in the sense that the cost and time of communication between two points approximates that of any two points chosen at random. Enterprise networks are often, not to say usually, designed and intended to be as flat as possible.

It is time to abandon the flat network. Flat networks lower the cost of attack against a network of systems or applications—successfully attacking a single node gains access to the network. Secure and trusted communication must now trump ease of any-to-any communication.

It is time for end-to-end encryptions for all applications. Think TLS, VPNs, VLANs and physically segmented networks. Encrypted pathways must reach all the way to applications or services and not stop at network perimeters or operating systems. Software Defined Networks put this within the budget of most enterprises.

Q: Most file systems use the old Unix convention of regulating access by the read-write-execute bits. Why is this a security problem and what would be a better practice for controlling access?

A: It is not so much a question of the controls provided by the file system but the permissive default policy chosen by management. It is a problem because it makes us vulnerable to data leakage, system compromise, extortion, ransomware, and sabotage. It places convenience and openness ahead of security and accountability. It reduces the cost of attack to that of duping an otherwise unprivileged user into clicking on a bait object.

It is time to abandon this convenient but dangerously permissive default access control rule of in favor of the more restrictive "read/execute-only" or even better, "Least privilege." These rules are more expensive to administer but they are more effective; they raise the cost of attack and shrink the population of people who can do harm. Our current strategies of convenience over security and "ship low-quality early and patch late" are proving to be not just ineffective and inefficient, but dangerous. They are more expensive in maintenance and breaches than we could ever have imagined.

The most efficient measures are those that operate early, preventing the malware from being installed and executed in the first place.

Q: What about malware? When it gets on your computer it can do all sorts of harm such as stealing your personal data or in the worst case ransomware. What effective defenses are there against these attacks?

A: The most efficient measures are those that operate early, preventing the malware from being installed and executed in the first place. This includes familiar antivirus programs as well as the restrictive access control rules mentioned earlier. It may include explicitly permitting only intended code to run (so-called "white listing"). It will include process-to-process isolation, which prevents malicious code from spreading; isolation can be implemented at the operating system layer, as in for example, Apple's iOS, or failing that, by running the untrusted processes in separate hardware boxes. We should not be running vulnerable applications such as email and browsing on porous operating systems, such as Windows and Linux, along with sensitive enterprise applications.

However, since prevention will never be much more than 80% effective, we should also be monitoring for indicators of compromise, the evidence of its presence that any code, malicious or otherwise, must leave.

Oh, I almost forgot. We must monitor traffic flows. Malware generates anomalous and unexpected traffic. Automated logging and monitoring of the origin and destination of all traffic moves from "nice to do" to "must do." While effective logging generates large quantities of data, there is software to help in the efficient organization and analysis of this data.

Q: Early in the development of operating systems we looked for solutions to the problem of running untrusted software on our computers. The principle of confinement was very important. The idea was to execute the program in a restricted memory where it could not access any data other than that which it asked for and which you approved. The basic von Neumann architecture did not have anything built in that would allow confinement. The modern operating systems like iOS or Android include confinement functions called "sandboxes" to protect users from untrusted software downloaded from the Internet. Is this a productive direction for OS designers and chip makers?

A: The brilliance of the von Neumann architecture was that it used the same storage for both procedures and data. While this was convenient and efficient, it is at the root of many of our current security problems. It permits procedures to be contaminated by their data and by other procedures, notably malware. Moreover, in a world in which one can put two terabytes of storage in one's pocket for less than $100, the problem that von Neumann set out to solve—efficiently using storage—no longer exists.

In the modern world of ubiquitous and sensitive applications running in a single environment, with organized criminals and hostile nation-states, convenience and efficiency can no longer be allowed to trump security. It is time to at least consider abandoning the open and flexible von Neumann Architecture for closed application-only operating environments, like Apple's iOS or the IBM iSeries, with strongly typed objects and APIs, process-to-process isolation, and a trusted computing base (TCB) protected from other processes. These changes must be made in the architecture and operating systems. There is nothing the iOS user can do from the user interface that will make a persistent change to the integrity of the software. There is little the developers of programs can do that will nullify defects in the operating system or other programs.

It is ironic that one can get a so-called "computer science" degree without even being aware of alternatives to the von Neumann architecture.

Q: There have been many attempts at intrusion detection in operating systems. Is it possible to identify that someone appearing to be an authorized user is actually someone else?

A: There are recognizable differences in the behavior of authorized users and impersonators. The simple measure of identifying repeated failed attempts to do something can reveal intruders. More complex measures exploiting advances in artificial intelligence can detect more subtle differences. We must tune these measures to balance false positives against the failure to detect. We must also ensure the alarms and alerts get to the responsible managers, usually the manager of the user and the owner of the asset, who are in a position to recognize the need for, and have the authority and resources, to take any indicated corrective action.

Q: When OSs started to span networks, traffic analysis of packets became an ingredient of a signature of computer use. Is this a valuable approach today?

A: It's tough but not hopeless. While we may never be sure that all nodes in the public networks properly identify themselves, cryptography can improve the trust that we have as to the source of traffic. While we may never solve the problem of compromised systems being used as "cut-outs" to hide the identity and location of the sources of attack traffic, by storing more meta data about the sources and destination of traffic, we can improve the effectiveness and efficiency of forensics.

Q: Another common attack method is phishing: email or voicemail messages that appear legitimate and entice you into revealing your personal information. Are there any practical ways to defend against phishing.

Outsiders may damage the brand but insiders may bring down the business.

A: Courtney's Third Law taught us "there are management solutions to technical problems but there are no technical solutions to management problems." Substitute "human" for "management" and the statement remains true.

Masquerading and fraud attacks appeal to the Seven Deadly Sins and to gullibility, fear, curiosity, and even the mere desire to be helpful. Fraud and deceit—what the roque hackers call "social engineering"—are as old as language. They have exploited every communication medium ever used.

However, in the modern world, these appeals are mostly used to get us to compromise our credentials or the integrity of our systems. We can caution and train our users but experience suggests the best of these efforts will not be sufficient. We must also use the measures recommended here to limit the consequences of the inevitable errors.

Q: What about insider attacks?

A: Threats have both source and rate. Insiders have a low rate but high consequences. Outsiders may damage the brand but insiders may bring down the business.

There are risks with privileged users and escalation of privileges. Edward Snowden was able to expand his privileges in an organization with "security" in its name. He did this over an extended period of time without being detected.

Pervasively we have too many over privileged users, with too little accountability. Indeed privileged users are among the most likely to share IDs and passwords. There is no accountability if something goes wrong. Often the privileges are so great and accountability so poor that the privileges, once granted, cannot be reliably withdrawn.

To reduce this threat, start with strong authentication for the use of any privileged capabilities. Implement multiparty controls over these capabilities. Improve accountability by ensuring privilege is available to only one user at a time, only when needed. Keep a record of all grants and uses of privilege.

Q: You clearly have strong opinions about how to secure our computer systems and networks. You place a great deal of weight on past security practices. Are these not obsolete? Don't we need the results of modern security research more than ever?

A: I plead guilty to having strong opinions and I beg for tolerance. I would like to defend my respect for past practices. Believe it or not, designers of operating systems have made security and protection a high priority since the 1960s. Their research and experience with real systems proves that many of the methods they discovered work. It astounds me that we would downplay those older successes in favor of unproven research.

What has changed over those years is not the need for security, but the risks and costs of insecurity. It should be clear to a casual reader of the news, let alone those with access to intelligence sources, that what we are doing is not working. It is both costly and dangerous.

While these recommendations may represent a change in the way we are doing things, we know they work. There is little new in them. Most of these ideas are as old as computing and some we inherited from more primitive information technology. Most of the resistance to using these practices comes from loss of convenience. Good security is not convenient. But it is absolutely necessary for the security of our assets and the reliability of the many critical systems on which we all depend. We need not suffer from the scourge of systems that so easily succumb to invaders.

Peter J. Denning (pjd@nps.edu) is Distinguished Professor of Computer Science and Director of the Cebrowski Institute for information innovation at the Naval Postgraduate School in Monterey, CA, is Editor of ACM Ubiquity, and is a past president of ACM. The author's views expressed here are not necessarily those of his employer or the U.S. federal government.

DOI:10.1145/3306617

Sepehr Vakil and Jennifer Higgs

▸ Mark Guzdial, Column Editor

Education
It's About Power

A call to rethink ethics and equity in computing education.

THIS COLUMN AIMS to build on and extend the field's understandings of the nature of ethics and equity in computing. Specifically, we argue that issues related to systems of power, which are often absent from conversations around ethics in computing, must be brought to the foreground in K–16 computing education. To this end, we argue for a justice-centered pedagogy[5] that centers power by explicitly acknowledging the ethical and political dimensions of computation and builds learning conditions so that everyone—including, but not limited to, students on computer science (CS) or engineering pathways—can understand, analyze, critique, and reimagine the technologies that shape everyday lives.

A power-conscious approach to ethics in computing highlights the sociopolitical and sociocultural contexts in which technologies are developed and deployed. To respond to the highly complex sociotechnical problems of the 21st century and beyond, future computer scientists and engineers need educational opportunities that prepare them to understand and care about the far-reaching ethical and sociopolitical implications of new technologies. Yet, we must also fundamentally rethink *who computing education is for.* Serious efforts should be made at the K–12 and undergraduate levels to make the knowledge, skills, and tools to critically examine the relationships between power, ethics, and technology available to all. Given rapidly evolving innovations and contexts of computing, we argue for two changes in our approach to ethics and equity in K–16 computing education:

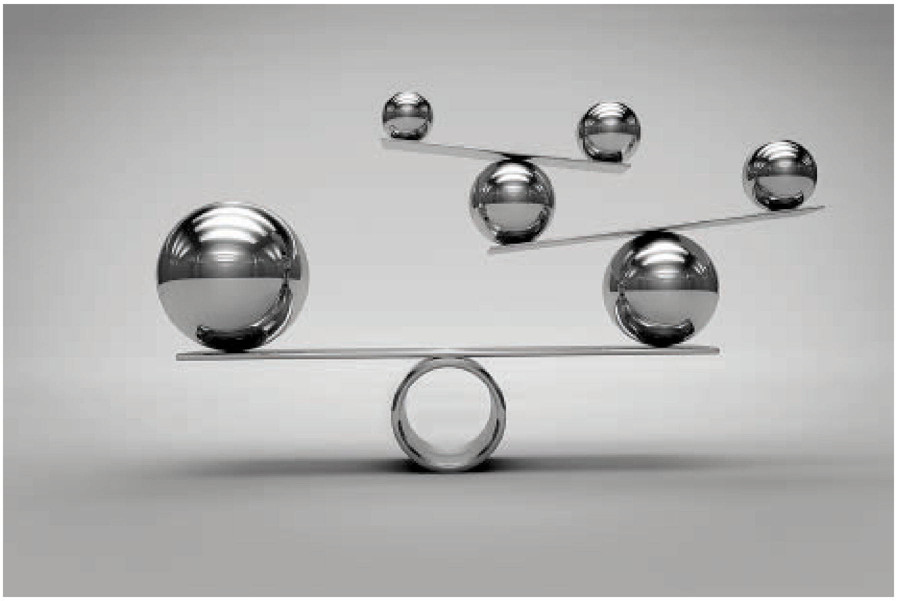


IMAGE FROM SHUTTERSTOCK.COM

- We must center power in discussions of ethics in computing, by which we mean explicitly attending to how computing systems intersect with structures of inequality and hierarchy in society; and
- We must view engagement with the sociopolitical and ethical dimensions of computing as a core practice made available to *all* students, whether or not they are on CS or engineering pathways.

Equity Is More than Inclusion

In recent years, the role of equity in CS education has increasingly become a topic of discussion. Much of this dialogue has centered around the creation of inclusive learning environments in computing, particularly with regard to marginalized students and their communities.[3] Yet, often missing from these well-intentioned conversations has been a robust consideration of equity in CS as it pertains to issues of ethics and power. In particular, the ways in which computational tools and technologies have multiple, complex, and profound implications for the lived experiences of nondominant communities have been largely ignored (for example, how machine learning is changing law enforcement practices in communities of color, how automation technologies are reshaping welfare eligibility,[1] or how commercial search engines reinforce racist and sexist bias).[4] Leaving these power imbalances unexamined precludes deep engagement with issues of equity. In our view, because these complicated interactions of technologies and society shape how nondominant groups experience and negotiate daily life and broader social systems, substantive discussions of equity in CS must intentionally include dynamics of power and ethics.

A Reframing of Ethics in Computing

While there have been a number of important calls and initiatives to integrate ethics into computing education, the tendency has been to ignore how ethics are situated within larger political and ideological contexts. As a result, discussions of ethics are primarily framed as a matter of personal choice and responsibility. For example, the current ACM Code of Ethics and Professional Conduct notes principles such as "Be honest and trustworthy" and "Know and respect existing rules pertaining to professional work." We have no bone to pick with universally accepted traits such as honesty and respect, but we contend that organizing discussions of ethics around the good or bad decisions/values of individual actors obscures more complex interactions between ethics and technology.

Moreover, an honest assessment of ethical behavior (for individuals as well as systems) must include analysis of how people's behaviors contribute to, resist, or otherwise intersect with structures of inequality and hierarchy in society. For example, say an engineer works at a firm where she is instructed to write code that programs handheld helmet-mounted imaging systems designed for the military. The engineer does her job faithfully as an honest, hard-working employee. Her code is elegant, original, and well documented. Yet, by helping to produce this slick and sophisticated technology, she also contributes to the project of militarism around the world. Is she acting ethically? Or we might ask: How do broader ethical and ideological values guide innovation in companies like the one this engineer works for? Does the current and emerging landscape of new technologies (and the institutions and industries creating these technologies) collectively contribute to a more just and ethical society? Centering power in discussions of ethics does not mean answers to these questions are provided for students, but it does mean opportunities are intentionally created for students to discuss, debate, and analyze what others have called the "macroethics" of technological systems.[2]

Organizing discussions of ethics around the good or bad decisions/values of individual actors obscures more complex interactions between ethics and technology.

A Focus on Decoding Power

A focus on power entails providing opportunities for students to decode how computational systems, which we define as coordinated networks of digital tools and devices (for example, the Internet, blockchain technology, surveillance systems), intersect and are intertwined with sociopolitical systems (for example, racism, neoliberalism, militarism, the U.S. immigration system). Decoding requires careful study of these different systems and the ways in which they interact. An unprecedented level of public debate recently has underscored the urgency of attending to these intersections in discussions of ethics and computing. How does racial bias shape artificial intelligence (AI) algorithms? How do theoretical advances in cryptography lay the foundation for mass surveillance? Why are engineers at Google and Microsoft raising concerns about their companies' entanglements with the Pentagon and Immigration and Customs Enforcement (ICE)? Addressing these highly complex questions requires a deeper understanding of how these technological systems interact with sociopolitical systems. For example, exploring racial bias in AI algorithms demands an understanding of visual cognition systems *and* systems of race and hierarchy. Developing a moral stance on war-related technologies, and evaluating those of others, requires understanding not just how technologies may be used for unethical purposes, but also how the politics of war and empire shape the technologies that are developed in the first place. These are fraught intersections, where ethical dilemmas arise and thrive; where

technology and society collide to simultaneously create challenges and opportunities for education and social action.

A Critical Practice for Democracy and Civic Engagement

Focusing on power in discussions of computing and ethics foregrounds justice and equity, and is thus a critical practice that can benefit all members of society. Democratic societies are shaped, filtered, enhanced, and circumscribed by computing technologies and the algorithms driving them, yet these interactions between society and technology are often difficult to discern. Full social and political participation hinges on the ability to perceive and interrogate these interactions. Today's and tomorrow's civically engaged actors must have access to technology and opportunities to develop technical skills, but they must also possess the knowledge, conceptual frameworks, and vocabularies to make sense of, vote, protest, design, and advocate for socially desirable configurations between society and technology. Centering power in considerations of ethics prepares people to foreground how various forms of injustice may be disputed or reproduced when considering interactions between technology and society.

A Commitment to Traversing Disciplinary Boundaries

Engaging the ethics and politics of computing demands an unprecedented and vigorous transdisciplinary dialogue between CS and the social sciences and humanities. Computer science instructors will need to move beyond decontextualized modules on ethics or individual courses on social impact that deemphasize moral and political questions. Universities will need to create learning pathways where students gain knowledge and skills to build the technologies of the future as they simultaneously develop the sensibilities and intellectual integrity to question, modify, or reimagine these technologies.

Toward these ends, there are encouraging cross-disciplinary developments on the horizon the field should support and continue to foster. Several universities with highly ranked CS programs are expanding CS learning opportunities in interesting ways (for instance, Northwestern's joint Ph.D. program in Computer Science and the Learning Sciences, and the new interdisciplinary College of Computing at MIT). The digital social sciences and humanities have started to examine the intersections of computational tools and methods in fields such as history, literature, film studies, political science, philosophy, and sociology. Liberal arts colleges are beginning to introduce technology requirements and offer specializations in areas such as artificial intelligence and data science. Much of this work aims to unite computational and humanistic questions in novel ways and inspire new ways of seeing and thinking about computation and its place in our society and lives. In middle and secondary computer science education, however, ethical and political dimensions of computing tend to be sidelined, including within introductory courses such as Exploring Computer Science (ECS) or CS Principles.[5] A pedagogical focus on power and ethics in K–12 CS education has the exciting potential to forge new disciplinary bridges between the goals and practices of CS and parallel efforts to engage youth in civics and social justice. Additionally, intentionally broadening the intellectual and social purposes of CS could invite a wider range of student identities.

There are encouraging cross-disciplinary developments on the horizon the field should support and continue to foster.

History as Our Guide

For computing education as a field to rethink ethics and equity in ways called for here will undoubtedly require a hard (and perhaps uncomfortable) epistemological and pedagogical pivot. We would do well, though, to remember a rich intellectual history of thinkers in our field who have laid a foundation upon which we may build. For instance, mathematician, philosopher, and pacifist Norbert Wiener forwarded a view of ethics rooted in the fundamental relationships between science and power. Especially in his later writings, he urged the field to take seriously the ways machines may alter society in ways that would challenge the very meaning of human life.[6] More recently, Jeannette Wing's contention that computational thinking is "a universally applicable attitude and skill set [that] everyone, not just computer scientists" can learn and use[7] helped spark an enduring debate about computation's transdisciplinarity and its untapped potential to inspire new ways of seeing the world. We see much value in these early formulations, particularly with regard to their emphasis on the *power of computing* to transform society. Highlighting power as a conceptual and pedagogical approach locates learning about computing within a justice frame that both complements and challenges previously articulated visions for computing education.

Robust understandings of power, ethics, equity, technologies, and society—as called for in this column—are key for the design of future tools and artifacts rooted in deep notions of the public good and social welfare. Future generations must possess the ability to critically analyze the affordances and constraints of technological advancement, as well as the moral imagination and technical skill to create with compassion and ethical integrity.

References

1. Eubanks, V. *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor*. St. Martin's Press, New York, NY, 2018.
2. Herkert, J.R. Ways of thinking about and teaching ethical problem solving: Microethics and macroethics in engineering. *Science and Engineering Ethics 11*, 3 (Mar. 2005), 373–385.
3. Margolis, J. *Stuck in the Shallow End: Education, Race, and Computing*. MIT Press, Boston, MA, 2010.
4. Noble, S.U. *Algorithms of Oppression: How Search Engines Reinforce Racism*. NYU Press, NY, 2018.
5. Vakil, S. Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education. *Harvard Educational Review 88*, 1 (Jan. 2018), 26–52.
6. Wiener, N. Some moral and technical consequences of automation. *Science 131*, 3410 (1960), 1355–1358.
7. Wing, J.M. Computational thinking. *Commun. ACM 49*, 3 (Mar. 2006), 33–35.

Sepehr Vakil (sepehr.vakil@northwestern.edu) is Assistant Professor, Learning Sciences, in the School of Education and Social Policy at Northwestern University, Evanston, IL, USA.

Jennifer Higgs (jmhiggs@ucdavis.edu) is Assistant Professor, Learning & Mind Sciences and Language, Literacy, & Culture, in the School of Education at the University of California, Davis, CA, USA.

DOI:10.1145/3265747

Mike Tissenbaum, Josh Sheldon, and Hal Abelson

Viewpoint
From Computational Thinking to Computational Action

Envisioning computing education that both teaches and empowers.

COMPUTATIONAL ACTION, A new framing for computing education, proposes that while learning about computing, young people should also have opportunities to create with computing that have direct impact on their lives and their communities. In this Viewpoint, we outline two key dimensions of computational action—computational identity and digital empowerment—and further argue that by focusing on computational action in addition to computational thinking, we can make computing education more inclusive, motivating, and empowering for young learners. Learners have the capacity to develop computational products that can have authentic impact in their lives from the moment they begin learning to code, all they need is to be situated in contexts that allow them to do so.

Too often, K–12 computing education has been driven by an emphasis on kids learning the "fundamentals" of programming. Even more progressive CS education that centers around the development of learners' computational thinking has largely focused on learners understanding the nuanced elements of computation, such as variables, loops, conditionals, parallelism, operators, and data handling.[10] This initial focus on the concepts and processes of computing, leaving real-world applications for "later" runs the risk of making learners feel that computing is not important for them to learn. It begs the question far too many math or physics students have asked, "When will we use this in our lives?"[1]

While there have been attempts to situate computing education in real-world contexts and problems, they are often generic (for example, designing checkout systems for supermarkets) and fail to connect to the specific personal interests and lives of learners. Though real-world application of their work is valuable for all learners, not providing opportunities to develop computational solutions with real-world potential is particularly problematic for young women and youth from nondominant groups. For these groups, who have been traditionally underrepresented in the computing fields, it has been observed that a sense of fitting into and belonging to the broader computing community is closely tied to being able to develop computational solutions that matter to themselves and those in their communities.[8] By connecting with students' real lives, we can help them develop a critical consciousness of

IMAGE BY BOYKO PICTURES

the role they can play in affecting their communities through computing and empower them to move beyond simply learning to code. Instead, we can ask them what they want to code and why they want to code it.[5]

By situating computing education in real-world contexts that matter to students, we can engage more people in computing, with all the benefits that affords the youth and to society. Though this may help to produce much-needed programmers, it will also produce computationally literate, problem-solving citizens.

Reducing the Barriers for Putting Computational Action into Practice

There are many challenges young learners face when trying to develop impactful computational solutions. Many of these can be attributed to the context of computing education itself—often taking place in traditional computing labs, which are far removed from students' everyday lives. However, with the growing proliferation of mobile and ubiquitous computing, there is the potential for rethinking and recontextualizing where and how students learn computing. Computing education can now be freed from the desk-bound screen and connected to students' lives and communities.

The ability to connect to the lives of students represents a fundamental shift in computing, opening up new avenues for young people to see their worlds as "possibility spaces," spaces in which they can ask questions and build solutions that address personally identified needs. However, in order to empower young people to build these solutions, we need to provide platforms and learning environments that reduce the barriers for them to quickly build and implement their designs. As one example, we developed App Inventor, a blocks-based programming language that allows learners to build fully functional mobile applications without the need to deal with complicated syntax.

Computational Action: A New Way of Framing Computing Education for Impact

The fundamental shift in the role computing can play in students' lives also requires us to critically reexamine the goals of CS education, particularly for K–12 students. The goal of computing education needs to move beyond computational thinking to a perspective of *computational action*. A computational action perspective on computing is founded on the idea that, while learning about computing, young people should have the opportunity to do computing in ways that have direct impact on their lives and their communities.

This fundamental shift in the role computing can play in students' lives also requires us to critically reexamine the goals of CS education.

Through multiple design studies, workshops, and global mobile app development initiatives that used MIT App Inventor, we have developed two key dimensions for understanding and developing educational experiences that support students in engaging in computational action: computational identity and digital empowerment. *Computational identity* builds on prior research that showed the importance of young people's development of scientific identity for future STEM growth.[6] For us, computational identity is a person's recognition of themselves as capable of designing and implementing computational solutions to self-identified problems or opportunities. Further, the students should see themselves as part of a larger community of computational creators. *Digital empowerment* builds from the work of Freire[2], which situates empowerment as the ability to critically engage in issues of concern to them, and Thomas and Velthouse,[9] who see empowerment connecting to the concepts of meaningfulness, competence, self-determination, and impact. As such, digital empowerment involves instilling in young learners the belief they can put their computational identity into action in authentic and meaningful ways on issues that matter to them.

In order to develop computational action educational initiatives, we have developed a set of criteria that outline the critical elements required.

Supporting the formation of computational identity requires:

- Students must feel they are responsible for articulating and designing their solutions, rather than working toward predetermined "right" answers.
- Students need to feel their work is authentic to the practices and products of broader computing and engineering communities.

Supporting the formation of digital empowerment requires:

- A significant number of activities and development should be situated in contexts that are authentic and personally relevant.
- Students need to feel their work has the potential to make an impact in their own lives or their community.
- Students should feel they are capable of pursuing new computational opportunities as a result of their current work.

Computational Action in Action

We have seen firsthand the powerful effect a computational action approach can have to learning computer science. In the slums of Dharavi in Mumbai (one of the largest slums in Asia, and the iconic location of the film *Slumdog Millionaire*), a group of young women (8–16 years old) recognized women's safety was a critical problem in their community. Despite having no prior programming experience, they were driven by the feeling they could effect real change in the lives of those close to them. Through guidance from a local mentor, some online videos, and MIT's App Inventor, they were able to build the Women Fight Back app, which focuses on women's safety and has features like SMS alerts, location mapping, distress alarm, and emergency calls to contacts.[4] Inspired by their early success, these young women built several more apps, including one to coordinate water pickup from public water sources, and an educational app for girls who cannot go to school. These young learners' growth from no computing experience to a group that is continually working to improve their community through computing, shows the transformational potential computational action can have.

Building on the success of the Dharavi girls and other young learners like them, we have begun developing formal computing curricula that incorporate the computational action model. Recently, working with teachers at a large, extremely diverse, urban, U.S. high school, we created a 10-week computing curriculum with App Inventor. In this curriculum, students developed computing solutions to an issue that was personally relevant and meaningful to them and their community: raising awareness and cleaning up the local riverway. Exit interviews highlighted positive changes in the students' perceptions of their own computational identities and digital empowerment. From not believing themselves capable of building mobile apps at all, they realized they could not only build apps, but that their designs could have significant real-world impact. Many students also expressed excitement to build new apps in the future.

Facilitating this kind of learner-driven and action-focused computing education requires a reexamination of how we provide support for learners. It also poses new challenges for teachers. Students need scaffolding in the design process to help them understand how to decompose their apps into manageable and buildable parts. Importantly, teachers need to be comfortable in complex, real-world situations that do not have a predefined solution. While this should not require teachers to learn more about programming functionally, it will require them to be more flexible in how it is applied. It will require new strategies for helping students discover solutions on their own (rather than giving them the answer), and it will require new ways of assessing student work. Recognizing these pedagogical shifts means we must embrace new educational approaches as we test and refine our theories on computational action.

Learners Recognize Opportunities to Apply Computing, then Design and Build Solutions

Having students drive their learning or problem-solving process is not a new idea in education. Problem-based learning (see for example Hmelo-Silver[3]) has been increasingly used in science and engineering education over the past two decades. However, putting the products students design into their communities has been a persistent challenge. Through the proliferation of mobile and ubiquitous computing, we are beginning to realize this potential.

By focusing on computational action instead of computational thinking, we engage kids in meaningful projects rather than canned exercises.

By focusing on computational action instead of computational thinking, we engage kids in meaningful projects rather than canned exercises. Papert argued that in the process of developing personally meaningful projects, students would be able to forge ideas and would learn the necessary coding elements by addressing challenges as they naturally arise.[7] This is similar to how much programming and computational solution building works in the professional world. People from all occupations and avocations alike come up with "projects" they want to build for which computer programs are necessary. These people plan ahead and begin building, but inevitably, obstacles arise. These computer programmers, professionals and amateurs, computer scientists, engineers, scientists, and many others, find answers to those problems within the broader community of programmers (by asking colleagues directly or through sites such as StackOverflow). If this is the how computing happens in the real world, why is the educational system so often focused on students learning computing and computational problem solving in abstracted and inauthentic ways?

With rapid changes happening in both computing and computing education landscapes, we have an opportunity to reconsider how students learn computing. Young learners have the capacity to develop computational products that have authentic impact in their lives from the moment they begin to code. They simply need contexts that allow them to have such impact. Computational action starts to define what these contexts should look like. With more computing instructors coming online, we have a unique opportunity to work with them as they develop skills and practices necessary to engage in computational action with their students. We are excited about a world in which young learners see the world as full of opportunities for them to digitally create the future they (and we) want to inhabit.

References

1. Flegg, J., Mallet, D., and Lupton, M. Students' perceptions of the relevance of mathematics in engineering. *Intl. Journal of Mathematical Education in Science and Technology 43*, 6 (June 2012), 717–732.
2. Freire, P. *Pedagogy of the Oppressed* (20th anniversary ed.). Continuum, NY, 1993.
3. Hmelo-Silver, C.E. Problem-based learning: What and how do students learn? *Educational Psychology Review 16*, 3 (Mar. 2004), 235–266.
4. Joshi, S. Teenage girl coders from Mumbai slum are building apps to solve local problems. (Mar. 29, 2016); http://mashable.com/2016/03/29/mumbai-dharavi-girls-coding-apps/
5. Lee, C.H. and Soep, E. None but ourselves can free our minds: Critical computational literacy as a pedagogy of resistance. *Equity & Excellence in Education 49*, 4 (Apr. 2016), 480–492.
6. Maltese, A.V. and Tai, R.H. Eyeballs in the fridge: Sources of early interest in science. *International Journal of Science Education 32*, 5 (May 2010), 669–685.
7. Papert, S. An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning 1*, 1 (Jan. 1996), 95–123.
8. Pinkard, N. et al. Digital youth divas: Exploring narrative-driven curriculum to spark middle school girls' interest in computational activities. *Journal of the Learning Sciences 26*, 3 (Mar. 2017); doi.org/10.1080/10508406.2017.1307199
9. Thomas, K.W. and Velthouse, B.A. Cognitive elements of empowerment: An "interpretive" model of intrinsic task motivation. *Academy of Management Review 15*, 4 (Apr. 1990), 666–681.
10. Wing, J.M. Computational thinking. *Commun. ACM 49*, 3 (Mar. 2006), 33–35.

Mike Tissenbaum (miketissenbaum@gmail.com) is Assistant Professor in the College of Education at the University of Illinois at Urbana-Champaign, IL, USA.

Josh Sheldon (jsheldon@mit.edu) is Associate Director, App Inventor, at MIT, Cambridge, MA, USA.

Hal Abelson (hal@mit.edu) is Class of 1922 Professor of Computer Science and Engineering in the Department of Electrical Engineering and Computer Science at MIT. Cambridge, MA, USA.

ACM Transactions on
Data Science

DOI:10.1145/3303868

Article development led by acmqueue
queue.acm.org

Blockchain remains a mystery, despite its growing acceptance.

BY JIM WALDO

A Hitchhiker's Guide to the Blockchain Universe

IT IS DIFFICULT these days to avoid hearing about blockchain. Blockchain is going to be the foundation of a new business world based on smart contracts. It is going to allow everyone to trace the provenance of their food, the parts in the items they buy, or the ideas they hear. It will change the way we work, the way the economy runs, and the way we live in general.

Despite the significant potential of blockchain, it is also difficult to find a consistent description of what it really is. A recent Google search for "blockchain technical papers" returned nothing but white papers for the first three screens; not a single paper is peer-reviewed. One of the best discussions of the technology itself is from the National Institute of Standards and Technology, but at 50-plus pages, it is a bit much for a quick read.[9]

The purpose of this article is to look at the basics of blockchain: the individual components, how those components fit together, and what changes might be made to solve some of the problems with blockchain technology. This technology is far from monolithic; some of the techniques can be used (at surprising savings of resources and effort) if other parts are cut away.

Because there is no single set of technical specifications, some systems that claim to be blockchain instances will differ from the system described here. Much of this description is taken from the original blockchain paper.[6] While details may differ, the main ideas stay the same.

Goals of Blockchain

The original objective of the blockchain system was to support "an electronic payment system based on cryptographic proof instead of trust ..."[6] While the scope of use has grown considerably, the basic goals and requirements have remained consistent.

The first of these goals is to ensure the anonymity of blockchain's users. This is accomplished by use of a public/private key pair, in a fashion that is reasonably well known and not reinvented by the blockchain technology. Each participant is identified by the public key, and authentication is accomplished through signing with the private key. Since this is not specific to blockchain, it is not considered further here.

The second goal is to provide a public record or *ledger* of a set of transactions that cannot be altered once verified and agreed to. This was originally designed to keep users of electronic currency from double-spending and to allow public audit of all transactions. The ledger is a record of what transactions have taken place, and the order of those transactions. The use of this ledger for verification of transactions other than the exchange of electronic cash has been the main extension of the blockchain technology.

The final core goal is for the system

A Hitchhiker's Guide to the Blockchain Universe

to be independent of any central or trusted authority. This is meant to be a peer- or participant-driven system in which no entity has more or less authority or trust than any other. The design seeks to ensure the other goals as long as more than half of the members of the participating community are honest.

Components of Blockchain

While there are lots of different ways to implement a blockchain, all have three major components. The first of these is the ledger, which is the series of blocks that are the public record of the transactions and the order of those transactions. Second is the consensus protocol, which allows all of the members of the community to agree on the values stored in the ledger. Finally, there is the digital currency, which acts as a reward for those willing to do the work of advancing the ledger. These components work together to provide a system that has the properties of stability, irrefutability, and distribution of trust that are the goals of the system.

The ledger is a sequence of blocks, where each block is an ordered sequence of transactions of an agreed-upon size (although the actual size varies from system to system). The first entry into a block is a cryptographic hash (such as those produced by the Secure Hash Algorithm SHA-256) of the previous block. This prevents the contents of the previous block from being changed, as any such change will alter the cryptographic hash of that block and thus can be detected by the community. These hash functions are easy to compute but (at least to our current knowledge) impossible to reverse. So once the hash of the contents of a block is published, anyone in the community can easily check that the hash is correct.

So far, this is nothing new; it is simply a Merkle chain, which has been in use for years. The wrinkle in blockchain is that the calculation of the hash needs to add a nonce (some random set of bits) to the block being hashed until the resulting hash has a certain number (generally six or eight) of leading zeros. Since there is no way to predict the value that will give that number of leading zeros to the hash, this is a brute-force calculation, which is exponentially difficult on the number of zeros required. This makes the calculation of the hash for the block computationally difficult and means any member of the community has the chance of coming up with an acceptable hash with a probability that is proportional to the amount of computing resources the member throws at the problem.

IMAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

Coming up with the hash and the right nonce is a proof of work (and, perhaps, luck) that can be easily verified by anyone in the community. Those attempting to calculate the right hash value for a block are the *miners* of the blockchain world; they are exchanging computation for pay.

Once a miner comes up with the right nonce that produces the right hash, they broadcast the result to the rest of the community, and all miners start work on the next block. The first entry in the new block will be the hash of the last block, and the second entry in the block will be the creation of some amount of currency assigned to the miner who found the hash for the previous block.

This works only if you have a block to start the chain. This is done in the same way all systems get started: by cheating and declaring a block to be the Genesis block.

It is possible that two different miners could both find, at the same time (or close enough), a nonce that gives a candidate hash value with the right number of leading zeros, or that someone seeing a nonce that works could claim the discovery as their own. There could even be two different blocks being proposed as the next entry in the chain. Dealing with such issues requires the next component of the system: the consensus protocol.

Consensus protocols are among the most-studied aspects of distributed systems. While it was proved some time ago that no algorithm will guarantee consensus if there is a possibility of any kind of failure,[3] a number of well-known protocols such as Paxos[4] have been used in systems for some time to give highly reliable mechanisms for distributed agreement. In consensus protocols such as Paxos, however, it is assumed the systems that must reach agreement are known.

Depending on the failure model used, the number of systems that must agree to reach consensus changes. When a majority of systems agree in such a protocol (for some definition of majority), consensus has been reached in systems that want to protect from non-byzantine failure. If the system is subject to byzantine failure, then two-thirds of the systems (plus one) need to agree. While the voting can be done in peer-to-peer systems, most efficient versions of the algorithms depend on a leader to initiate the voting and tally the results.

In the blockchain universe, however, there is a trust-free system, which means there can be no leader. Further, in the blockchain universe the number of systems participating in validating the transactions (that is, finding a hash for the block with the right number of zeros in the prefix) is not known. This makes claims that a block is accepted when 51% of the miners agree on the block nonsense, since there is no known value for the number of entities trying to agree.

Instead, the majority is determined by the calculation of the hash for the next block. Since that block begins with the hash of the previous block, and since the likelihood of the next block's hash being calculated is proportional to the amount of computing resources trying to calculate the appropriate hash for the next block, if a majority of the computing power available to the miners starts to work on a block that is seeded with the previous hash, then that block is more likely to be offered as the next block. This is the reason for consensus being tied to the longest chain, as that chain will be produced by the largest number of computing resources.

This mechanism relies on the generation of a hash with the right set of leading zeros being genuinely random. Being random also means that on occasion someone will get lucky and a chain that is being worked on by a minority of the miners will be hashed appropriately before a chain that is being worked on by a larger amount of computing resources.

In an important sense, however, this does not matter. The blockchain universe defines a majority as the production of an appropriate nonce and hash. Sometimes this means more than half of the computing power has worked on the problem, but other times it might mean only one (exceptionally lucky) miner got the answer. This might mean a set of transactions in a block that is not verified first need to be rolled back, but that is the nature of in-flight transactions.

It does mean all of the miners in the blockchain universe need to move to a newly hashed block as the basis for the calculation of the next block in the chain. This requires an incentive mechanism, which is where the third component of the blockchain universe enters the picture: digital currency.

Digital currency. The reason for a miner to do all the computational work to calculate the nonce and hash of a block is that the first to do so gets an allocation of digital currency as the first transaction in the next block. This also encourages other miners to accept a block as quickly as possible, so they can start doing the work to hash the next block (which has likely been filled with transactions during the time it took to hash the previous block). Bitcoin was the original blockchain currency and incentive; in September 2017 the reward for hashing a block was 12 bitcoins[8] when the exchange rate was 1 bitcoin = ~$4,500 U.S. (prices fluctuate rather wildly). This reward halves (for bitcoin) every 210,000 blocks. The next halving is expected around May 25, 2020.[1]

Other digital currencies work in a similar fashion. To spend the currency, entries are made in the then-current block, which acts as a ledger of all the currency exchanges for a particular ledger/digital coin combination.

Problems with Blockchain

While blockchain was originally proposed as a mechanism for trustless digital currency, its uses have expanded well beyond that particular use case. Indeed, the emphasis seems to have bifurcated into companies that emphasize the original use for currency (thus the explosion of initial coin offerings, which create new currencies) and the use of the ledger as a general mechanism for recording and ordering transactions. For the first use, the claim is that blockchain can replace outdated notions of currency and allow a new, private, friction-free economy. For the latter use, the claim is that blockchain can be used to track supply chains, create self-enforcing contracts, and generally eliminate layers of mediation in any transaction.

Both of these kinds of uses present some serious problems. Many are problems any new technology encounters in replacing entrenched interests, but a number of them are technical in nature; those are the ones discussed here.

A number of criticisms of blockchain center on the mechanism used to create an accepted hash for a block. To ensure this can be discovered by anyone, the mechanism needs to be one that takes significant computation but can be easily verified. To ensure the blocks that are verified cannot be changed, the computation needs to be impractical to reverse. Hashing the block using a function such as SHA-256 and requiring that a nonce value is added until some number of leading zeros appears in the hash fits these characteristics nicely. This very set of requirements, however, means the consensus mechanism has intrinsic limitations.

Scaling. An obvious worry about the consensus-by-hashing mechanism used in blockchain is whether the technology can scale to the levels needed for more general use. According to blockchain.com, the number of confirmed transactions averages around 275,000 per day, with a peak over the last year of about 380,000.[2] This is an impressive number but hardly the 400,000 transactions per minute that major credit-card systems perform on peak days. Blocks can currently be verified at a rate of four to six per second, and this is the limiting factor on the number of transactions.

While there are a number of proposals to deal with scaling blockchain, it is unclear how these fit with the base design of the system. Making the verification of a block difficult and random is an important aspect of the basic design of blockchain; this is the proof of work that is at the core of the trustless consensus algorithm. If the verification of a block is made easier, then the probabilistic guarantees of any miner being able to discover the appropriate hash decreases, and the possibility of some miner with a large amount of computing taking over the chain increases. Verifying a block is meant to be hard; that's how the system avoids having to trust any particular member or set of members.

One mechanism suggested for scaling is to shard the blockchain into a number of different chains, so that transactions can be done in parallel in different chains. This is happening in the different coin exchanges; each coin system can be thought of as a separate shard. This introduces its own complexity in order to have a transaction that crosses these shards, since the notion of ensured consistency requires that all ledgers are self-contained to allow consistency checking within each ledger. A new blockchain could be created to be used for cross-blockchain transactions, but the incentive mechanism for that blockchain would be a new electronic currency that would need to stay within the ecosystem of this new blockchain. Getting the interacting blockchains to trust the mediating blockchain is an unsolved problem.

While blockchain was originally proposed as a mechanism for trustless digital currency, its uses have expanded well beyond that particular use case.

There have also been attempts to use some mechanism other than proof of work to drive the consensus protocol. Perhaps the best known of these is the proof-of-stake approach, in which a block can be calculated in much simpler ways, and consensus is reached when those with a majority of the currency agree on the hashing of the block. Since the amount of currency and its owners are known, this is not subject to the problem of not knowing the members of the community to vote. But this does reintroduce the notion of trust to the system; those who have more money have more of a stake, and therefore are trusted more than those who have less of a stake. This is the electronic equivalent of an oligarchy, which has not worked particularly well in the past but might prove more stable in this context.

Power consumption. A second criticism of blockchain technology that is an outgrowth of the consensus mechanism is the amount of energy consumed in the discovery of an appropriate hash for a block. Calculating a hash with the appropriate number of leading zeros requires many hashing calculations, which in turn burn a lot of electricity; some have claimed that bitcoin and related cryptocurrencies are mechanisms to transform electricity into currency. The estimates of how much electricity is consumed range from the low side stating that it is about as much as is used by the city of San Jose, CA, to the high side that it is equivalent to Denmark's power consumption. No matter which model is used for the calculation, the answer is large.

The hope is that this energy drain will diminish, perhaps by changing the hardware used for the hashing to something far more efficient (such as specialized ASICs). Making the hashing process more efficient, however, is at odds with blockchain's fundamental mechanism of trusting no one; the point is that the verification of a block must be difficult and random so that any miner is equally likely to find the hash.

The energy consumption might be less worrisome if the calculations eating all of this power were generally useful. SETI@home, for example, uses a considerable amount of energy by offloading analysis of background radio-wave transmissions to Internet-connected computers. This initiative, based at UC Berkeley's SETI (Search for Extraterrestrial Intelligence) Research Center, is trying to find signs of other intelligent life in the universe, which is seen by the participants as worth doing (and paying for the extra electricity).

Perhaps the calculation used to verify the blockchain could be changed to something that offered more than just verification of the blockchain. Such a calculation would need to have the properties of being equally possible for all miners to find (given equality of computing resource), difficult to find, and easy to verify. It is not clear what this calculation might be.

Trust. Perhaps the most problematic aspect of blockchain is its core notion of being trustless. Much of the complexity of the technology is caused by this requirement. It is unclear, however, that this is even necessary for the kinds of uses people talk about as core to blockchain, or that the system is actually free of trust.

It is because of the lack of trust that the system requires verification of the block to be computationally difficult, one-way, and easy to verify. If this requirement of trustlessness were dropped, then production of a public ledger that was unchangeable and easily verified could be done easily. Suppose such a ledger is to be used for inter-bank transfer (which has been suggested as a use for blockchain). Instead of a trustless system, however, the users decide to trust a consortium of major banks, the Federal Reserve Board, and some selection of consumer watchdog agencies or organizations. This consortium could choose a member (perhaps on a rotating basis) who is responsible for keeping the ledger (a leader). Transactions are written to the ledger, and when the ledger block reaches an appropriate size, the leader hashes the ledger, uses the hash to start a new block, and continues (just as in the current blockchain).

The difference is that there is no need for the leader to randomly try values added to the block until the right number of leading zeros is produced in the hash. Without that requirement, the hash can be done very quickly with little energy expense. The block still can't be changed (since the hash is still a one-way function), and any member of the consortium (or anyone else who has access to the ledger) can quickly check the hash. A public, verifiable, and unchangeable ledger can be produced in this way but at much lower cost in both time and energy.

This does require trust in the various members of the consortium, but verifying that the consortium is not cheating on the hashing of a block would be easy. This is not a fully centralized trust in a single entity but, rather, trusting a group. The larger and more varied the group, the less likely the group would collude. Note also that such a system does not need an incentive mechanism such as a digital currency to operate.

Who Do You Trust?

Maybe you really do not want to trust anyone. Calibrating paranoia is difficult, and perhaps you really do want to have an economic system in which no specifiable set of entities has the ability to collude and control the system. That is the real reason for blockchain.

As Ken Thompson pointed out in 1984, trust has to happen somewhere.[7] Even if you do not trust any group to calculate the blocks, you need to trust the developers of the software being used to manage the blocks, the ledgers, and the rest. Everything from bugs to design changes[5] in the software have led to forks in the bitcoin ecosystem that have caused considerable churn in those systems. If your trust is in the security and solidity of the code, that is a choice you make. But it is not a trustless system.

A public, nonrefutable, unalterable ledger for transactions could be a useful tool for a number of applications. Building such a system on top of known cryptographic protocols could be done in a number of ways. Doing it on top of a system such as blockchain is needed if the requirement that the system be trustless (except for trusting the software) is added. Such a trustless system comes with a cost.

Whether the cost is worth it is a decision that requires an understanding of the various parts of the system and how they interact. A public, unforgeable, unchangeable ledger is possible without cryptocurrency or a consensus algorithm based on a difficult-to-compute one-way function that is easily verified. Cryptocurrencies can be created without the use of either a public ledger or a trustless consensus algorithm. And consensus algorithms can be created that do not require a financial incentive system or a public ledger.

Related articles on queue.acm.org

Bitcoin's Academic Pedigree
Arvind Narayanan and Jeremy Clark
https://queue.acm.org/detail.cfm?id=3136559

Research for Practice: Cryptocurrencies, Blockchains, and Smart Contracts; Hardware for Deep Learning
https://queue.acm.org/detail.cfm?id=3043967

Certificate Transparency
Ben Laurie, Google
https://queue.acm.org/detail.cfm?id=2668154

References

1. Bitcoinblockhalf.com. Bitcoin block reward halving countdown.
2. Blockchain.com. Confirmed transactions per day, 2018; https://www.blockchain.com/charts/n-transactions?daysAverageString=7.
3. Fischer, M., Lynch, N.A., Paterson, M. Impossibility of distributed consensus with one faulty process. *JACM 32*,2 (1985), 374–382.
4. Lamport, L. The part-time parliament. *ACM Trans. Computer Systems 16*, 2 (1998), 133–169.
5. Morris, D.Z. Bitcoin is in wild upheaval after the cancellation of the Segwit2x fork. *Fortune* (Nov. 12, 2017); http://fortune.com/2017/11/12/bitcoin-upheavel-segwit2x-fork/.
6. Nakamoto, S. Bitcoin, a peer-to-peer electronic cash system, 2008; https://bitcoin.org/bitcoin.pdf.
7. Thompson, K. Reflections on trusting trust. *Commun. ACM 27*, 8 (Aug. 1984), 761–763; https://dl.acm.org/citation.cfm?id=358210.
8. Trubetskoy, G. Electricity cost of 1 bitcoin (Sept. 2017); https://grisha.org/blog/2017/09/28/electricity-cost-of-1-bitcoin/.
9. Yaga, D., Mell, P., Roby, N., Scarfone, K. Blockchain technology overview. NISTIR 8202 (Oct. 2018). National Institute of Standards and Technology; https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf.

Jim Waldo is a professor of the practice of computer science at Harvard University, where he is also the chief technology officer for the School of Engineering, a position he assumed after leaving Sun Microsystems Laboratories.

DOI:10.1145/3303878

Article development led by acmqueue
queue.acm.org

Four challenging work situations and how to handle them.

BY KATE MATSUDAIRA

Design Patterns for Managing Up

HAVE YOU EVER been in a situation where you are presenting to your manager or your manager's manager and you completely flub the opportunity by saying all the wrong things? Me too. It is from such encounters that I started to put together design patterns for handling these difficult situations. I like to think in systems and patterns, so applying this way of thinking to communication just makes sense. I have also found these rules of thumb are useful to others, so I would like to share them here.

When you can spot the patterns, you can use some of the ideas presented here as guidelines to navigate these tricky, high-stress scenarios. This way you can feel confident and capable as a leader because you will know what to do: how to solve the problem and what steps to follow next.

Here are some of the most common challenging situations you may run into at work and how you can handle them.

1. Someone asks you something you don't know.

You are in a meeting (where you know you are expected to know all the answers) when someone asks a question of you, and you just aren't certain of the answer.

Sure, the obvious answer is to say, "I don't know." But what if the person asking you that question is a customer? What if that person is your VP? What if you are interviewing for a new role? Suddenly, the stakes are raised and you just don't want to say, "I don't know." You don't want to look uninformed or unprofessional. All of a sudden there is social pressure to be the person with the right answer.

In the moment, the desire to say anything but "I don't know" is so overwhelming that you may end up making up something on the spot, trying to be as vague as possible so you can't be wrong. But—as you know if someone has done this to you—it's usually obvious to the other people in the room that you don't know the answer, and now that interaction will negatively color their view of you.

They still end up knowing that you don't know, but now you have also wasted their time. Or even worse, you give them an answer you *think* is correct, but you turn out to be wrong. Then you are in a situation where you actually are uninformed, and you are left crossing your fingers hoping no one will ever find out.

If you take the long view, you will realize it is always much better to admit you don't know something, but then take action to fix it.

In this case the pattern is:

A. Admit you aren't certain.

B. Own the follow-up to determine the answer.

C. Give a timeline for when you will follow up.

D. Deliver a correct, concise, and thoughtful response.

When you say, "I don't know, but I can investigate and get back to you after lunch" or "I don't know, but ______ does and I will ask her and get back to you by the end of the day," suddenly you are a person solving a problem.

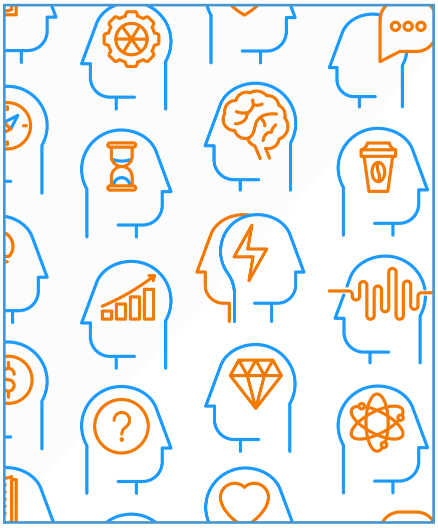
Any time you don't know, be clear that you don't know, but follow up with a plan for how you will get the information and with a deadline for conveying that information. That is all the person asking the question really wants anyway: for you to supply the answer.

Smart people don't know everything; they just know where to look to find out what they need to know.

2. There is a problem that is your fault or responsibility.

Have you ever been in a situation where something went wrong (such as a system outage) and you wanted to figure out the cause/solution to the problem before broadcasting it more widely? I know this is how I feel—especially if keeping that system up and running is my (or my team's) responsibility.

When problems occur, there is a natural instinct to hide or deny the problem is a problem, or that it is even happening at all. We want to minimize the problems that are our responsibility because, after all, a big part of our job is to make sure problems *don't* happen. When you are proactive about sharing and fixing a problem, however, it is actually an opportunity to show you are an asset to the bosses.



In general, it is always better to control the message and have your management learn of the problem from you (instead of a customer or a boss). Great leadership is keeping everyone on the same page, and it is your job to communicate proactively so there are no surprises.

If your system experiences an outage or any other customer-impacting issue, you should be the first person to share the issue with your manager, customers, team, or whoever is affected by the problem. This makes you the proactive one who discovered the problem and is already working to address it before anyone else even knows about it.

For this pattern the steps are:

A. Let the key people know you know about the problem and are working on a solution. Establish yourself as the owner and let them know you will see this through to the finish.

B. Share steps if you know them, but if you do not know the answer, let the key people know when you will provide the next update (for example, "We aren't sure what caused the problem, or the impact, but will provide another update in an hour").

C. Give a timeline. This is the most important thing you can do. When will the problem be fixed? If you don't know when or how it will be fixed yet, when will you provide an update? What possible solutions are you going to try and when will you know the results?

Give specific answers to specific questions. It is never okay just to say, "We're working on it."

Even if all you can say is that you are aware and will update with more information at a specific time, then you are at least able to fulfill a promise to your boss/team/customer by following through and giving them an update at the specified later time. This is far better than making them wait in radio silence while you try to figure out what is going on. They will spend that time feeling negative about you or your team, which only amplifies the impact of the problem.

Tip: Look for ways you can help your customers avoid having a bad experience with your product, whether with a notification letting them know about the problems or pointers to documentation to help them work around it.

3. There is a decision that you don't agree with.

When you care a lot about your work, it can be really difficult to get behind a decision you don't agree with. You might feel like it is worth speaking up—but this can be challenging when that decision is made by someone who is one, two, or more levels above you.

Sure, many technology execs claim they want everyone who disagrees to raise questions, but is that something you really want to gamble your career on? When it comes to navigating these situations, there is generally a right way and a wrong way to disagree.

When you first hear about a decision that you think is a bad call—such as a re-org or a new feature you believe is a waste of time—it can be a really fraught moment. You might feel frustrated or angry, which can lead you to give an overly emotional response that is both unproductive and ineffective.

Remember there is someone in the chain who thinks this is a good idea; that is why it is being implemented. So, it is worth your time to try to understand the "why" behind this idea—it just might change your mind, and even if it doesn't, it will give you the context you need to make an argument that could actually convince the decision maker that you are right.

Next time there is a decision that you don't agree with, instead of jumping to "no," try asking about the goal. What is this person trying to achieve? If you still don't agree with the explana-

IMAGE BY ALEXEY BLOGOODF

tion, you can try researching the problem and providing an alternative plan. (You will always be taken more seriously if you can provide another solution instead of just saying the current plan is a bad idea.)

Sometimes you will not change anyone's mind. If this is the case, then it's time to shift your perspective and implement the change as it has been outlined to you. Even so, there is still a benefit to understanding the reasoning behind a change you don't agree with.

As a leader, you do not want to tell your team the reason things are changing is "because ______ made me do it." That makes it seem like you have no power, as though you do things just because you are told to. It certainly won't inspire confidence in your team or make them any more likely to embrace the decision.

Instead, you want to explain that you are implementing a change because of XYZ reason. If questioned, you can mention you shared concerns about the plan with your leaders, but then reiterate the reasoning behind the current plan with your team. Ultimately, it is your job to commit to the organization's strategy and help your team succeed within that strategy.

There is never a 100% right answer at work. There are just different approaches and trade-offs with each approach. It's not your job to agree with your leadership 100% of the time, but it is your job to make the company as successful as possible.

So, when a decision is made that does not make sense to you, here are the steps:

A. Take the emotion out of it—wait a day or two if you need time to clear your head.

B. Don't disagree; ask about the context and reasons for the change.

C. Start with your manager or the main decision maker if you have a prior relationship, and then escalate up the chain of command together (rather than just emailing your thoughts to the CEO).

D. Research and present alternative options that will achieve the same goals.

E. If you do not succeed in swaying the decision maker, then support the plan of action. Be sure to share the context and reasoning with your team, and then do everything you can to make the situation better (which can be helping the cause succeed, mitigating fall-out, and so on).

4. Your manager gives you negative feedback.

I have a love/hate relationship with feedback. I love it because it is necessary to improve, but I hate it because no matter how much I try, I can't help but take it personally.

No matter how amazing you are at your job, you will sometimes get feedback about things you could be doing better. It can be difficult to hear, especially if you are someone who works really hard all the time. When it comes to negative feedback, it is important to reframe the conversation. Feedback isn't a bad thing. It is a gift, and you should always adopt a growth mind-set and see it as a chance to improve.

I once had a manager give me some feedback on a meeting I had been part of. The meeting had gone off the rails, and I had done my best to get people back on track and focused. My manager hadn't been there, but he had been told by other people in the meeting that my tactics had rubbed some people the wrong way.

In the moment, I was frustrated. I did not agree with the feedback at all, so I wanted to get to the bottom of it. This was a big mistake. Suddenly, my boss was on the defensive. I was asking him really direct questions: "What does that mean?" and "What could I have done differently?" I was forcing him to explain a situation he didn't really know about.

It harmed my relationship with that manager, and I had to do a lot of work to repair it.

I talked to my mentor about the situation, and my mentor reminded me that every time you get feedback it is an opportunity to grow. It is valuable, because it is a chance to learn more about how you can be better.

Keep in mind that as difficult as it is to receive negative feedback, it isn't always easy for your manager to give that feedback either. No one really enjoys conveying bad news. So, handling it the right way is also an opportunity to create a positive encounter with your manager. When you get feedback that hurts or that you do not agree with, try to remain calm in the moment. Focus on slowing down your breathing, be aware of your heart rate, and try to keep your face relaxed.

Then say, "I hear you. I will be more mindful of that in the future."

That's it, and the only pattern for this one.

If your manager has more to say, he or she will say it. If you are not clear on exactly what the person means or if you are genuinely interested in hearing suggestions for what you could do next time, ask. But don't snap at this person or start giving them the third degree about the feedback. If you think you might react emotionally or angrily, then simply say the phrase noted here, thank your boss for the feedback, and then leave.

You can always send an email message later, after you have had a chance to collect your thoughts, then get further clarification at that time. Let your boss know you have been thinking about it and took the feedback to heart. Ask in an open and authentic way for advice.

Look for patterns and be the version of yourself that you want to be. Challenges come up all the time at work. Spend time now thinking about how you want to be seen at work, and then think about how that version of you would respond to the challenges that you could encounter. When you have a plan in place, you are much more likely to succeed.

I hope your find these tips useful, and if you have additional ones to add to the mix, please leave a comment. C

Related articles on queue.acm.org

Views from the Top
Kate Matsudaira
https://queue.acm.org/detail.cfm?id=3156692

People and Process
James Champy
https://queue.acm.org/detail.cfm?id=1122687

Broken Builds
Kode Vicious
https://queue.acm.org/detail.cfm?id=1740550

Kate Matsudaira (katemats.com) is an experienced technology leader. She has worked at Microsoft and Amazon and successful startups before starting her own company, Popforms, which was acquired by Safari Books.

DOI:10.1145/3287287

Article development led by acmqueue
queue.acm.org

These attacks on statistical databases are no longer a theoretical danger.

BY SIMSON GARFINKEL, JOHN M. ABOWD, AND CHRISTIAN MARTINDALE

Understanding Database Reconstruction Attacks on Public Data

IN 2020, THE U.S. Census Bureau will conduct the Constitutionally mandated decennial Census of Population and Housing. Because a census involves collecting large amounts of private data under the promise of confidentiality, traditionally statistics are published only at high levels of aggregation. Published statistical tables are vulnerable to *database reconstruction attacks* (DRAs), in which the underlying microdata is recovered merely by finding a set of microdata that is consistent with the published statistical tabulations. A DRA can be performed by using the tables to create a set of mathematical constraints and then solving the resulting set of simultaneous equations. This article shows how such an attack can be addressed by adding noise to the published tabulations, so the reconstruction no longer results in the original data. This has implications for the 2020 census.

The goal of the census is to count every person once, and only once, and in the correct place. The results are used to fulfill the Constitutional requirement to apportion the seats in the U.S. House of Representatives among the states according to their respective numbers.

In addition to this primary purpose of the decennial census, the U.S. Congress has mandated many other uses for the data. For example, the U.S. Department of Justice uses block-by-block counts by race for enforcing the Voting Rights Act. More generally, the results of the decennial census, combined with other data, are used to help distribute more than $675 billion in federal funds to states and local organizations.

Beyond collecting and distributing data on U.S. citizens, the Census Bureau is also charged with protecting the privacy and confidentiality of survey responses. All census publications must uphold the confidentiality standard specified by Title 13, Section 9 of the U.S. Code, which states that Census Bureau publications are prohibited from identifying "the data furnished by any particular establishment or individual." This section prohibits the Census Bureau from publishing respondents' names, addresses, or any other information that might identify a specific person or establishment.

Upholding this confidentiality requirement frequently poses a challenge, because many statistics can inadvertently provide information in a way that can be attributed to a particular entity. For example, if a statistical agency *accurately* reports there are two persons living on a block and the average age of the block's residents is 35, that would constitute an improper disclosure of personal information, because one of the residents could look up the data, subtract their contribution, and infer the age of the other.

IMAGE BY IGOR STEVANOVIC

Of course, this is an extremely simple example. Statistical agencies have understood the risk of such unintended disclosure for decades and have developed a variety of techniques to protect data confidentiality while still publishing useful statistics. These techniques include *cell suppression*, which prohibits publishing statistical summaries from small groups of respondents; *top-coding*, in which ages higher than a certain limit are coded as that limit before statistics are computed; *noise-injection*, in which random values are added to some attributes; and *swapping*, in which some of the attributes of records representing different individuals or families are swapped. Together, these techniques are called statistical disclosure limitation (SDL).

Computer scientists started exploring the issue of statistical privacy in the 1970s with the increased availability of interactive query systems. The goal was to build a system that would allow users to make queries that would produce summary statistics without revealing information about individual records. Three approaches emerged: auditing database queries, so that users would be prevented from issuing queries that zeroed in on data from specific individuals; adding noise to the data stored within the database; and adding noise to query results.[1] Of these three, the approaches of adding noise proved to be easier because the complexity of auditing queries increased exponentially over time—and, in fact, was eventually shown to be NP (nondeterministic polynomial)-hard.[8] Although these results were all couched in the language of interactive query systems, they apply equally well to the activities of statistical agencies, with the *database* being the set of confidential survey responses, and the *queries* being the schedule of statistical tables that the agency intends to publish.

In 2003, Irit Dinur and Kobbi Nissim showed that it isn't even necessary for an attacker to construct queries on a database carefully to reveal its underlying confidential data.[4] Even a surprisingly small number of random queries can reveal confidential data, because the results of the queries can be combined and then used to "reconstruct" the underlying confidential data. Adding noise to either the database or to the results of the queries decreases the accuracy of the reconstruction, but it also decreases the accuracy of the queries. The challenge is to add sufficient noise in such a way that each individual's privacy is protected, but not so much noise that the utility of the database is ruined.

Subsequent publications[3,6] refined the idea of adding noise to published tables to protect the privacy of the individuals in the dataset. Then in 2006, Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith proposed a formal framework for understanding these results. Their paper, "Calibrating Noise to Sensitivity in Private Data Analysis,"[5] introduced the concept of *differ-*

ential privacy. They provided a mathematical definition of the privacy loss that persons suffer as a result of a data publication, and they proposed a mechanism for determining how much noise must be added for any given level of privacy protection (the authors received the Test of Time award at the Theory of Cryptography Conference in 2016 and the Gödel Prize in 2017).

The 2020 census is expected to count approximately 330 million people living on about 8.5 million blocks, with some inhabited blocks having as few as a single person and other blocks having thousands. With this level of scale and diversity, it is difficult to visualize how such a data release might be susceptible to database reconstruction. We now know, however, that reconstruction would in fact pose a significant threat to the confidentiality of the 2020 microdata that underlies unprotected statistical tables if privacy-protecting measures are not implemented. To help understand the importance of adopting formal privacy methods, this article presents a database reconstruction of a much smaller statistical publication: a hypothetical block containing seven people distributed over two households. (The 2010 U.S. Census contained 1,539,183 census blocks in the 50 states and the District of Columbia with between one and seven residents. The data can be downloaded from https://bit.ly/2L0Mk51)

Even a relatively small number of constraints results in an exact solution for the blocks' inhabitants. Differential privacy can protect the published data by creating uncertainty. Although readers may think that the reconstruction of a block with just seven people is an insignificant risk for the country as a whole, this attack can be performed for virtually every block in the United States using the data provided in the 2010 census. The final section of this article discusses the implications of this for the 2020 decennial census.

An Example Database Reconstruction Attack

To present the attack, let's consider the census of a fictional geographic frame (for example, a suburban block), conducted by the fictional statistical agency. For every block, the agency collects each resident's age, sex, and race, and publishes a variety of statistics. To simplify the example, this fictional world has only two races—black or African American, and white—and two sexes—female and male.

The statistical agency is prohibited from publishing the raw microdata and instead publishes a tabular report. Table 1 shows fictional statistical data for a fictional block published by the fictional statistics agency. The "statistic" column is for identification purposes only.

Notice that a substantial amount of information in Table 1 has been suppressed—marked with a (D). In this case, the statistical agency's disclosure-avoidance rules prohibit it from publishing statistics based on one or two people. This suppression rule is sometimes called "the rule of three," because cells in the report

Table 1. Fictional statistical data for a fictional block.

			Age	
Statistic	**Group**	**Count**	**Median**	**Mean**
1A	Total Population	7	30	38
2A	Female	4	30	33.5
2B	Male	3	30	44
2C	Black or African American	4	51	48.5
2D	White	3	24	24
3A	Single Adults	(D)	(D)	(D)
3B	Married Adults	4	51	54
4A	Black or African American Female	3	36	36.7
4B	Black or African American Male	(D)	(D)	(D)
4C	White Male	(D)	(D)	(D)
4D	White Female	(D)	(D)	(D)
5A	Persons Under 5 Years	(D)	(D)	(D)
5B	Persons Under 18 Years	(D)	(D)	(D)
5C	Persons 64 Years or Over	(D)	(D)	(D)

Note: Married persons must be 15 or over

Table 2. Possible ages for a median of 30 and a mean of 44.

A	B	C	A	B	C	A	B	C
1	30	101	11	30	91	21	30	81
2	30	100	12	30	90	22	30	80
3	30	99	13	30	89	23	30	79
4	30	98	14	30	88	24	30	78
5	30	97	15	30	87	25	30	77
6	30	96	16	30	86	26	30	76
7	30	95	17	30	85	27	30	75
8	30	94	18	30	84	28	30	74
9	30	93	19	30	83	29	30	73
10	30	92	20	30	82	30	30	72

Table 3. Variables associated with the reconstruction attack.

Person	Age	Sex	Race	Marital Status
1	A1	S1	R1	M1
2	A2	S2	R2	M2
3	A3	S3	R3	M3
4	A4	S4	R4	M4
5	A5	S5	R5	M5
6	A6	S6	R6	M6
7	A7	S7	R7	M7

Key			
Female	0		
Male	1		
Black or African American		0	
White		1	
Single			0
Married			1

sourced from fewer than three people are suppressed. In addition, complementary suppression has been applied to prevent subtraction attacks on the small cells.

Encoding the constraints. The database can be reconstructed by treating the attributes of the persons living on the block as a collection of variables. A set of constraints is then extracted from the published table. The database reconstruction finds a set of attributes that are consistent with the constraints. If the statistics are highly constraining, then there will be a single possible reconstruction, and the reconstructed microdata will necessarily be the same as the underlying microdata used to create the original statistical publication. Note that there must be at least one solution because the table is known to be formulated from a real database.

For example, statistic 2B states that three males live in the geography. This fictional statistical agency has previously published technical specifications that its computers internally represent each person's age as an integer. The oldest verified age of any human being was 122.[14] If we allow for unreported supercentenarians and consider 125 to the oldest possible age of a human being, there are only a finite number of possible age combinations, specifically:

$$\binom{125}{3} = \frac{125 \times 124 \times 123}{3 \times 2 \times 1} = 317{,}750$$

Within the 317,750 possible age combinations, however, there are only 30 combinations that satisfy the constraints of having a median of 30 and a mean of 44, as reported in Table 1. (Notice that the table does not depend on the oldest possible age, so long as it is 101 or over.) Applying the constraints imposed by the published statistical tables, the possible combinations of ages for the three males can be reduced from 317,750 to 30. Table 2 shows the 30 possible ages for which the median is 30 and the mean is 44.

To mount a full reconstruction attack, an attacker extracts all of these constraints and then creates a single mathematical model embodying all constraints. An automated solver can then find an assignment of the variables that satisfies these constraints.

SAT and SAT Solvers

The Boolean SAT problem was the first to be proven NP-complete.[9] This problem asks, for a given Boolean formula, whether replacing each variable with either true or false can make the formula evaluate to true. Modern SAT solvers work well and reasonably quickly in a variety of SAT problem instances and up to reasonably large instance sizes.

Many modern SAT solvers use a heuristic technique called CDCL (conflict-driven clause learning).[10] Briefly, a CDCL algorithm:

1. Assigns a value to a variable arbitrarily.

2. Uses this assignment to determine values for the other variables in the formula (a process known as unit propagation).

3. If a conflict is found, backtracks to the clause that made the conflict occur and undoes variable assignments made after that point.

4. Adds the negation of the conflict-causing clause as a new clause to the master formula and resumes from step 1.

This process is fast at solving SAT problems because adding conflicts as new clauses has the potential to avoid wasteful "repeated backtracks." Additionally, CDCL and its predecessor algorithm, DPLL (Davis–Putnam–Logemann–Loveland), are both provably complete algorithms: they will always return either a solution or "Unsatisfiable" if given enough time and memory. Another advantage is that CDCL solvers reuse past work when producing the universe of all possible solutions.

A wide variety of SAT solvers are available to the public for minimal or no cost. Although a SAT solver requires the user to translate the problem into Boolean formulae before use, programs such as Naoyuki Tamura's Sugar facilitate this process by translating user-input mathematical and English constraints into Boolean formulae automatically.

Sugar Input

Sugar input is given in a standard constraint satisfaction problem (CSP) file format. A constraint must be given on a single line of the file, but here we separate most constraints into multiple lines for readability. Constraint equations are separated by comments describing the statistics they encode.

Input for the model in this article is available at https://queue.acm.org/appendices/Garfinkel_SugarInput.txt.

To continue with the example, statistic 1A establishes the universe of the constraint system. Because the block contains seven people, and each has four attributes (age, sex, race, and marital status), that creates 28 variables, representing those four attributes for each person. These variables are A1... A7 (age), S1... S7 (sex), R1... R7 (race), and M1... M7 (marital status), as shown in Table 3. The table shows the variables associated with the DRA. The coding of the categorical attributes is presented in the key.

Because the mean age is 38, we know that:

$$A1 + A2 + A3 + A4 + A5 + A6 + A7 = 7 \times 38$$

The language Sugar[13] is used to encode the constraints in a form that can be processed by a SAT (satisfiability) solver. Sugar represents constraints as s-expressions.[11] For example, the age combination equation can be represented as:

```
; First define the integer
variables, with the range
0..125
(int A1 0 125)
(int A2 0 125)
(int A3 0 125)
(int A4 0 125)
(int A5 0 125)
(int A6 0 125)
(int A7 0 125)
; Statistic 1A: Mean age is 38
(= (+ A1 A2 A3 A4 A5 A6 A7)
  (* 7 38)
)
```

Once the constraints in the statistical table are turned into s-expressions, the SAT solver solves them with a brute-force algorithm. Essentially, the solver explores every possible combination of the variables, until a combination is found that satisfies the constraints. Using a variety of heuristics, SAT solvers are able to rapidly eliminate many combinations of variable assignments.

Despite their heuristic complexity,

SAT solvers can process only those systems that have Boolean variables, so Sugar transforms the s-expressions into a much larger set of Boolean constraints. For example, each age variable is encoded using unary notation as 126 Boolean variables. Using this notation, the decimal value 0 is encoded as 126 false Boolean variables, the decimal value 1 is encoded as 1 true and 125 false values, and so on. Although this conversion is not space efficient, it is fast, provided that the integers have a limited range.

To encode the median age constraint, the median of a group of numbers is precisely defined as the value of the middle number when the numbers are arranged in sorted order (for the case in which there is an odd number of numbers). Until now, persons 1 through 7 have not been distinguished in any way: the number labels are purely arbitrary. To make it easier to describe the median constraints, we can assert the labels must be assigned in order of age. This is done by introducing five constraints, which has the side effect of eliminating duplicate answers that have simply swapped records, an approach called *breaking symmetry*.[12]

```
(<= A1 A2)
(<= A2 A3)
(<= A3 A4)
(<= A4 A5)
(<= A6 A7)
```

Having the labels in chronological order, we can constrain the age of the person in the middle to be the median:

```
(= A4 30)
```

Sugar has an "if" function that allows encoding constraints for a subset of the population. Recall that statistic 2B contains three constraints: there are three males, their median age is 30, and their average age is 44. The value 0 represents a female, and 1 represents a male:

```
#define FEMALE 0
#define MALE 1
```

Using the variable Sn to represent the sex of person n, we then have the constraint:

```
S1 + S2 + S3 + S4 + S5 + S6 + S7 = 3
```

This can be represented as:

```
(= (+ S1 S2 S3 S4 S5 S6 S7) 3)
```

As illustrated in Figure 1, the if function allows a straightforward way to create a constraint for the mean age 44 of male persons.

Table 1 translates into 164 individual s-expressions extending over 457 lines. Sugar then translates this into a single Boolean formula consisting of 6,755 variables arranged in 252,575 clauses. This format is called the CNF (conjunctive normal form) because it consists of many clauses that are combined using the Boolean AND operation.

Interestingly, we can even create constraints for the suppressed data. Statistic 3A is suppressed, so we know

Figure 1. Encoding statistic 2B, that the average male age is 44, with Sugar's "if" statement.

```
(= (+ (if (= S1 MALE) A1 0) ; average male age = 44
      (if (= S2 MALE) A2 0)
      (if (= S3 MALE) A3 0)
      (if (= S4 MALE) A4 0)
      (if (= S5 MALE) A5 0)
      (if (= S6 MALE) A6 0)
      (if (= S7 MALE) A7 0)
      )
   (* 3 44))
```

Figure 2. Encoding the suppressed statistic 3A, that there are between 0 and 2 single adults.

Let Mn represent the marital status of person n:

```
#define SINGLE 0
#define MARRIED 1

(int SINGLE_ADULT_COUNT 0 2)
(= (+ (if (and (= M1 SINGLE) (> A1 17)) 1 0)
      (if (and (= M2 SINGLE) (> A2 17)) 1 0)
      (if (and (= M3 SINGLE) (> A3 17)) 1 0)
      (if (and (= M4 SINGLE) (> A4 17)) 1 0)
      (if (and (= M5 SINGLE) (> A5 17)) 1 0)
      (if (and (= M6 SINGLE) (> A6 17)) 1 0)
      (if (and (= M7 SINGLE) (> A7 17)) 1 0))
 SINGLE_ADULT_COUNT)

(>= SINGLE_ADULT_COUNT 0)
(<= SINGLE_ADULT_COUNT 2)
```

Table 4. A single satisfying assignment.

Age	Sex	Race	Marital Status	Solution #1
8	F	B	S	8FBS
18	M	W	S	18MWS
24	F	W	S	24FWS
30	M	W	M	30MWM
36	F	B	M	36FBM
66	F	B	M	66FBM
84	M	B	M	84MBM

Table 5. Solutions without statistic 4A.

Solution #1	Solution #2
8FBS	2FBS
18MWS	12MWS
24FWS	24FWM
30MWM	30MBM
36FBM	36FWS
66FBM	72FBM
84MBM	90MBM

that there are 0, 1, or 2 single adults, as no complementary suppression was required (see Figure 2).

Translating the constraints into CNF allows them to be solved using any solver that can solve an NP-complete program, such as a SAT solver, SMT (satisfiability module theories) solver, or MIP (mixed integer programming) solver. There are many such solvers, and most take input in the so-called DIMACS file format, which is a standardized form for representing CNF equations. The DIMACS format (named for the Center for Discrete Mathematics and Theoretical Computer Science at Rutgers University in New Jersey) was popularized by a series of annual SAT solver competitions. One of the results of these competitions was a tremendous speed-up of SAT solvers over the past two decades. Many solvers can now solve CNF systems with millions of variables and clauses in just a few minutes, although some problems do take much longer. Marijn Heule and Oliver Kullmann discussed the rapid advancement and use of SAT solvers in their 2017 article, "The Science of Brute Force."[7]

The open source PicoSAT[2] solver is able to find a solution to the CNF problem detailed here in approximately two seconds on a 2013 MacBook Pro with a 2.8GHz Intel i7 processor and 16GB of RAM (although the program is not limited by RAM), while the open source Glucose SAT solver can solve the problem in under 0.1 seconds on the same computer. The stark difference between the two solvers shows the speed-up possible with an improved solving algorithm.

Exploring the solution universe. PicoSAT finds a satisfying assignment for the 6,755 Boolean variables. After the solver runs, Sugar can translate these assignments back into integer values of the constructed variables. (SMT and MIP solvers can represent the constraints at a higher level of abstraction, but for our purposes a SAT solver is sufficient.)

There exists a solution universe of all the possible solutions to this set of constraints. If the solution universe contains a single possible solution, then the published statistics completely reveal the underlying confidential data—provided that noise was not added to either the microdata or the tabulations as a disclosure-avoidance mechanism. If there are multiple satisfying solutions, then any element (person) in common among all of the solutions is revealed. If the equations have no solution, either the set of published statistics is inconsistent with the fictional statistical agency's claim that it is tabulated from a real confidential database or an error was made in that tabulation. This doesn't mean that a high-quality reconstruction is not possible. Instead of using the published statistics as a set of constraints, they can be used as inputs to a multidimensional objective function: the system can then be solved to a set of variables as close as possible to the published statistics using another kind of solver called an optimizer.

The 2020 census is expected to count approximately 330 million people living on about 8.5 million blocks, with some inhabited blocks having as few as a single person and other blocks having thousands.

Normally SAT, SMT, and MIP solvers will stop when they find a single satisfying solution. One of the advantages of PicoSAT is that it can produce the solution universe of all possible solutions to the CNF problem. In this case, however, there is a single satisfying assignment that produces the statistics in Table 1. That assignment is seen in Table 4.

Table 1 provides some redundant constraints on the solution universe: some of the constraints can be dropped while preserving a unique solution. For example, dropping statistic 2A, 2B, 2C, or 2D still yields a single solution, but dropping 2A *and* 2B increases the solution universe to eight satisfying solutions. All of these solutions contain the reconstructed microdata records 8FBS, 36FBM, 66FBM, and 84MBM. This means that even if statistics 2A and 2B are suppressed, we can still infer that these four microdata records must be present.

Statistical agencies have long used suppression in an attempt to provide privacy to those whose attributes are present in the microdata; the statistics they typically drop are those based on a small number of people. How effective is this approach?

In Table 1, statistic 4A is an obvious candidate for suppression—especially given that statistics 4B, 4C, and 4D have already been suppressed to avoid an inappropriate statistical disclosure.

Removing the constraints for statistic 4A increases the number of solutions from one to two, shown in Table 5.

Defending Against a DRA

There are three approaches for defending against a database reconstruction attack. The first is to publish less statistical data—this is the approach taken by legacy disclosure-avoidance techniques (cell suppression, top-coding, and generalization). The second and third approaches involve adding noise, or randomness. Noise can be added to the statistical data being tabulated or to the results after tabulation. Each approach is considered here.

Option 1. Publish less data. Although it might seem that publishing less statistical data is a reasonable defense against the DRA, this choice may severely limit the number of tabulations that can be published. A related problem is that, with even a moderately small population, it may be computationally infeasible to determine when the published statistics still identify a sizable fraction of individuals in the population.

Option 2. Apply noise before tabulation. This approach is called *input noise injection*. For example, each respondent's age might be randomly altered by a small amount. Input noise injection does not prevent finding a set of microdata that is consistent with the published statistics, but it limits the value of the reconstructed microdata, since what is reconstructed is the microdata *after* the noise has been added.

For example, if a random offset in the range of 2... + 2 is added to each record of the census, and the reconstruction results in individuals of ages (7, 17, 22, 29, 36, 66, 82) or (6, 18, 26, 31, 34, 68, 82), an attacker would presumably take this into account but would have no way of knowing if the true age of the youngest person is 5, 6, 7, 8, or 9. Randomness could also be applied to the sex, race, and marital status variables. Clearly, the more noise that is added, the better privacy is protected, but the less accurate are the resulting statistics. Considering statistic 1A, input noise infusion might result in a median 28... 32 and a mean 36... 40. (Note that when using differential privacy, the infused noise is not drawn from a bounded domain but instead is typically drawn from a Laplace or geometric distribution.)

Swapping, the disclosure-avoidance approach used in the 2010 census, is a kind of input noise injection. In swapping, some of the attributes are exchanged, or *swapped*, between records. The advantage of swapping is that it has no impact on some kinds of statistics: if people are swapped only within a county, then any tabulation at the county level will be unaffected by swapping. The disadvantage of swapping is that it can have significant impact on statistics at lower levels of geography, and values that are not swapped are unprotected.

Statistical agencies have long used suppression in an attempt to provide privacy to those whose attributes are present on the microdata, although the statistics they typically drop are those based on a small number of people. How effective is this approach?

Option 3. Apply noise to the published statistics. This approach is called *output noise injection*. Whereas input noise injection applies noise to the microdata directly, output noise injection applies output to the statistical publications. Output noise injection complicates database reconstruction by eliminating naïve approaches based on the straightforward application of SAT solvers. Also, even if a set of microdata is constructed that is mostly consistent with the published statistics, these microdata will be somewhat different from the original microdata that was collected. The more noise that was added to the tabulation, the more the microdata will be different.

When noise is added to either the input data (option 2) or the tabulation results (option 3), with all records having equal probability of being altered, it is possible to mathematically describe the resulting privacy protection. This is the basis of differential privacy.

Implications for the 2020 census. The Census Bureau has announced that it is adopting a noise-injection mechanism based on differential privacy to provide privacy protection for the underlying microdata collected as part of the 2020 census. Following is the motivation for that decision.

The protection mechanism developed for the 2010 census was based on a swapping.[15] The swapping technique was not designed to protect the underlying data against a DRA. Indeed, it is the Census Bureau's policy that both the swapped and the unswapped microdata are considered confidential.

The 2010 census found a total population of 308,745,538. These people occupied 10,620,683 habitable blocks. Each person was located in a residential housing unit or institutional housing arrangement (what the Census Bureau calls "group quarters"). For each person, the Census Bureau tabulated the person's location, as well as sex, age, race, and ethnicity, and the person's relationship to the head of the household—that is, six attributes per person, for a total of approximately 1.5 billion attributes. Using this data, the Census Bureau published approximately 7.7 billion linearly independent statistics, including 2.7 billion in the PL94-171 redistricting file, 2.8 billion in the balance of summary file 1, 2 billion in summary file 2, and 31 million records in a public-use microdata sample. This results in approximately 25 statistics per person. Given these numbers and the example in this article, it is clear that there is a theoretical possibility the national-level census could be reconstructed, although tools such as Sugar and PicoSAT are probably not powerful enough to do so.

To protect the privacy of census respondents, the Census Bureau is developing a privacy-protection system based on differential privacy. This system will ensure every statistic and the corresponding microdata receive some amount of privacy protection, while providing that the resulting statistics are sufficiently accurate for their intended purpose.

This article has explained the motivation for the decision to use differential privacy. Without a privacy-protection system based on noise injection, it would be possible to reconstruct accurate microdata using only the published statistics. By using differential privacy, we can add the minimum amount of noise necessary to achieve the Census Bureau's privacy requirements. A future article will explain how that system works.

Related Work

In 2003, Irit Dinur and Kobbi Nissim[4] showed the amount of noise that must be added to a database to prevent a reconstruction of the underlying data is on the order of $\Omega(\sqrt{n})$ where n is the number of bits in the database. In practice, many statistical agencies do not add this much noise when they release statistical tables. (In our example, each record contains 11 bits of data, so the confidential database has 77 bits of information. Each statistic in Table 3 can be modeled as a four-bit of count, a seven-bit of median, and a seven-bit of mean, for a total of 18 bits; Table 3 releases 126 bits of information.) Dinur and Nissim's primary finding is that many statistical agencies leave themselves open to the risk of database reconstruction. This article demonstrates one way to conduct that attack.

Statistical tables create the possibility of database reconstruction because they form a set of constraints for which there is ultimately only one exact solution when the published table is correctly tabulated from a real confidential database. Restricting the number or specific types of queries—for example, by suppressing results from a small number of respondents—is often insufficient to prevent access to indirectly identifying information, because the system's refusal to answer a "dangerous" query itself provides the attacker with information.

Conclusion

With the dramatic improvement in both computer speeds and the efficiency of SAT and other NP-hard solvers in the last decade, DRAs on statistical databases are no longer just a theoretical danger. The vast quantity of data products published by statistical agencies each year may give a determined attacker more than enough information to reconstruct some or all of a target database and breach the privacy of millions of people. Traditional disclosure-avoidance techniques are not designed to protect against this kind of attack.

Faced with the threat of database reconstruction, statistical agencies have two choices: they can either publish dramatically less information or use some kind of noise injection. Agencies can use differential privacy to determine the minimum amount of noise necessary to add, and the most efficient way to add that noise, in order to achieve their privacy protection goals.

Acknowledgments

Robert Ashmead, Chris Clifton, Kobbi Nissim, and Philip Leclerc provided extraordinarily useful comments on this article. Naoyuki Tamura provided invaluable help regarding the use of Sugar.

Related articles on queue.acm.org

Go Static or Go Home
Paul Vixie
https://queue.acm.org/detail.cfm?ref=rss&id=2721993

Privacy, Anonymity, and Big Data in the Social Sciences
Jon P. Daries et al.
https://queue.acm.org/detail.cfm?id=2661641

Research for Practice: Private Online Communication; Highlights in Systems Verification
Albert Kwon, James Wilcox
https://queue.acm.org/detail.cfm?id=3149411

References

1. Adam, N.R., Worthmann, J.C. Security-control methods for statistical databases: A comparative study. *ACM Computing Surveys 21*, 4 (1989), 515–556; http://doi.acm.org/10.1145/76894.76895.
2. Biere, A. PicoSAT essentials. J. Satisfiability, *Boolean Modeling and Computation 4*, (2008), 75–97; https://bit.ly/2QczizW
3. Blum, A., Dwork, C., McSherry, F. Nissim, K. Practical privacy: The SuLQ framework. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2005, 128–138; https://dl.acm.org/citation.cfm?id=1065184.
4. Dinur, I., Nissim, K. Revealing information while preserving privacy. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Principles of Database Systems*, 2003, 202–210; https://dl.acm.org/citation.cfm?id=773173.
5. Dwork, C., McSherry, F., Nissim, K., Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, 2006, 265–284. Springer-Verlag, Berlin, Heidelberg; http://dx.doi.org/10.1007/11681878_14.
6. Dwork, C., Nissim, K. Privacy-preserving datamining on vertically partitioned databases. In *Proceedings of the 24th International Cryptology Conference* 3152, 2004, 528–544. Springer Verlag, Santa Barbara, CA; https://bit.ly/2zKunmJ
7. Heule, M.J.H., Kullmann, O. The science of brute force. *Commun. ACM 60*, 8 (Aug. 2017), 70–79; http://doi.acm.org/10.1145/3107239.
8. Kleinberg, J., Papadimitriou, C., Raghavan, P. Auditing Boolean attributes. In *Proceedings of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2000, 86–91; http://doi.acm.org/10.1145/335168.335210.
9. Kong, S., Malec, D. Cook-Levin theorem. Lecture, University of Wisconsin, 2007.
10. Marques-Silva, J., Lynce, I., Malik, S. Conflict-driven clause learning SAT solvers. *Handbook of Satisfiability*, 131–153. IOS Press, Amsterdam, The Netherlands, 2009.
11. McCarthy, J. Recursive functions of symbolic expressions and their computation by machine, part I. *Commun. ACM 3*, 4 (Apr. 1960), 184–195; https://dl.acm.org/citation.cfm?id=367199.
12. Metin, H., Baarir, S., Colange, M., Kordon, F. CDCLSym: Introducing effective symmetry breaking in SAT solving. *Tools and Algorithms for the Construction and Analysis of Systems.* D. Beyer and M. Huisman, eds. Springer International Publishing, 2018, 99–114; https://link.springer.com/chapter/10.1007/978-3-319-89960-2_6.
13. Tamura, N., Taga, A., Kitagawa, S., Banbara, M. Compiling finite linear CSP into SAT. *Contraints 14*, 2 (2009), 254–272; https://dl.acm.org/citation.cfm?id=1527316; http://bach.istc.kobe-u.ac.jp/sugar/.
14. Whitney, C.R. Jeanne Calment, world's elder, dies at 122. *New York Times* (Aug. 5, 1997); https://nyti.ms/2kM4oFb.
15. Zayatz, L., Lucero, J., Massell, P., Ramanayake, A. Disclosure avoidance for Census 2010 and American Community Survey five-year tabular data products. Statistical Research Division, U.S. Census Bureau; https://www.census.gov/srd/CDAR/rrs2009-10_ACS_5yr.pdf.

Simson L. Garfinkel is the Senior Computer Scientist for Confidentiality and Data Access at the U.S. Census Bureau and the Chair of the Census Bureau's Disclosure Review Board.

John M. Abowd is the Chief Scientist and Associate Director for Research and Methodology at the U.S. Census Bureau, where he serves on leave from his position as the Edmund Ezra Day Professor of Economics, professor of information science, and member of the Department of Statistical Sciences at Cornell University, Ithaca, NY, USA.

Christian Martindale is a senior at Duke University, Durham, NC, USA.

DOI:10.1145/3241036

The kind of causal inference seen in natural human thought can be "algorithmitized" to help produce human-level machine intelligence.

BY JUDEA PEARL

The Seven Tools of Causal Inference, with Reflections on Machine Learning

THE DRAMATIC SUCCESS in machine learning has led to an explosion of artificial intelligence (AI) applications and increasing expectations for autonomous systems that exhibit human-level intelligence. These expectations have, however, met with fundamental obstacles that cut across many application areas. One such obstacle is adaptability, or robustness. Machine learning researchers have noted current systems lack the ability to recognize or react to new circumstances they have not been specifically programmed or trained for. Intensive theoretical and experimental efforts toward "transfer learning," "domain adaptation," and "lifelong learning"[4] are reflective of this obstacle.

Another obstacle is "explainability," or that "machine learning models remain mostly black boxes"[26] unable to explain the reasons behind their predictions or recommendations, thus eroding users' trust and impeding diagnosis and repair; see Hutson[8] and Marcus.[11] A third obstacle concerns the lack of understanding of cause-effect connections. This hallmark of human cognition[10,23] is, in my view, a necessary (though not sufficient) ingredient for achieving human-level intelligence. This ingredient should allow computer systems to choreograph a parsimonious and modular representation of their environment, interrogate that representation, distort it through acts of imagination, and finally answer "What if?" kinds of questions. Examples include interventional questions: "What if I make it happen?" and retrospective or explanatory questions: "What if I had acted differently?" or "What if my flight had not been late?" Such questions cannot be articulated, let alone answered by systems that operate in purely statistical mode, as do most learning machines today. In this article, I show that all three obstacles can be overcome using causal modeling tools, in particular, causal diagrams and their associated logic. Central to the development of these tools are advances in graphical and structural models that have made counterfactuals computationally manageable and thus rendered causal reasoning a viable com-

» **key insights**

- **Data science is a two-body problem, connecting data and reality, including the forces behind the data.**
- **Data science is the art of interpreting reality in the light of data, not a mirror through which data sees itself from different angles.**
- **The ladder of causation is the double helix of causal thinking, defining what can and cannot be learned about actions and about worlds that could have been.**

ILLUSTRATION BY VAULT49

ponent in support of strong AI.

In the next section, I describe a three-level hierarchy that restricts and governs inferences in causal reasoning. The final section summarizes how traditional impediments are circumvented through modern tools of causal inference. In particular, I present seven tasks that are beyond the reach of "associational" learning systems and have been (and can be) accomplished only through the tools of causal modeling.

The Three-Level Causal Hierarchy

A useful insight brought to light through the theory of causal models is the classification of causal information in terms of the kind of questions each class is capable of answering. The classification forms a three-level hierarchy in the sense that questions at level i (i = 1, 2, 3) can be answered only if information from level j ($j > i$) is available.

Figure 1 outlines the three-level hierarchy, together with the characteristic questions that can be answered at each level. I call the levels 1. Association, 2. Intervention, and 3. Counterfactual, to match their usage. I call the first level Association because it invokes purely statistical relationships, defined by the naked data.[a] For instance, observing a customer who buys toothpaste makes it more likely that this customer will also buy floss; such associations can be inferred directly from the observed data using standard conditional probabilities and conditional expectation.[15] Questions at this layer, because they require no causal information, are placed at the bottom level in the hierarchy. Answering them is the hallmark of current machine learning methods.[4] The second level, Intervention, ranks higher than Association because it involves not just seeing what is but changing what we see. A typical question at this level would be: What will happen if we double the price? Such a question cannot be answered from sales data alone, as it involves a change in customers' choices in reaction to the new pricing. These choices may differ substantially from those taken in previous price-raising situations—unless we replicate precisely the market conditions that existed when the price reached double its current value. Finally, the top level invokes Counterfactuals, a mode of reasoning that goes back to the philosophers David Hume and John Stuart Mill and that has been given computer-friendly semantics in the past two decades.[1,18] A typical question in the counterfactual category is: "What if I had acted differently?" thus necessitating retrospective reasoning.

I place Counterfactuals at the top of the hierarchy because they subsume interventional and associational questions. If we have a model that can answer counterfactual queries, we can also answer questions about interventions and observations. For example, the interventional question: What will happen if we double the price? can be answered by asking the counterfactual question: What would happen had the price been twice its current value? Likewise, associational questions can be answered once we answer interventional questions; we simply ignore the action part and let observations take over. The translation does not work in the opposite direction. Interventional questions cannot be answered from purely observational information, from statistical data alone. No counterfactual question involving retrospection can be answered from purely interventional information, as with that acquired from controlled experiments; we cannot re-run an experiment on human subjects who were treated with a drug and see how they might behave had they not been given the drug. The hierarchy is therefore directional, with the top level being the most powerful one.

Counterfactuals are the building blocks of scientific thinking, as well as of legal and moral reasoning. For example, in civil court, a defendant is considered responsible for an injury if, but for the defendant's action, it is more likely than not the injury would not have occurred. The computational meaning of "but for" calls for comparing the real world to an alternative world in which the defendant's action did not take place.

Each layer in the hierarchy has a syntactic signature that characterizes the sentences admitted into that layer. For example, the Association layer is characterized by conditional probability sentences, as in $P(y|x) = p$, stating that: The probability of event $Y = y$, given that we observed event $X = x$ is equal to p. In large systems, such evidentiary sentences can be computed efficiently through Bayesian networks or any number of machine learning techniques.

At the Intervention layer, we deal with sentences of the type $P(y|do(x), z)$ that denote "The probability of event $Y = y$, given that we intervene and set the value of X to x and subsequently observe event $Z = z$. Such expressions can be estimated experimentally from randomized trials or analytically using causal Bayesian networks.[18] A child learns the effects of interventions through playful manipulation of the environment (usually in a deterministic playground), and AI planners obtain interventional knowledge by exercising admissible sets of actions. Interventional expressions cannot be inferred from passive observations alone, regardless of how big the data.

Finally, at the Counterfactual level, we deal with expressions of the type $P(yx\,|x',y')$ that stand for "The probability that event $Y = y$ would be observed had X been x, given that we actually observed X to be x' and Y to be y'." For example, the probability that Joe's salary would be y had he finished college, given that his actual salary is y' and that he had only two years of college." Such sentences can be computed only when the model is based on functional relations or structural.[18]

This three-level hierarchy, and the formal restrictions it entails, explains why machine learning systems, based only on associations, are prevented from reasoning about (novel) actions, experiments, and causal explanations.[b]

[a] Other terms used in connection with this layer include "model-free," "model-blind," "black-box," and "data-centric"; Darwiche[5] used "function-fitting," as it amounts to fitting data by a complex function defined by a neural network architecture.

[b] One could be tempted to argue that deep learning is not merely "curve fitting" because it attempts to minimize "overfit," through, say, sample-splitting cross-validation, as opposed to maximizing "fit." Unfortunately, the theoretical barriers that separate the three layers in the hierarchy tell us the nature of our objective function does not matter. As long as our system optimizes some property of the observed data, however noble or sophisticated, while making no reference to the world outside the data, we are back to level-1 of the hierarchy, with all the limitations this level entails.

Questions Answered with a Causal Model

Consider the following five questions:

- How effective is a given treatment in preventing a disease?;
- Was it the new tax break that caused our sales to go up?;
- What annual health-care costs are attributed to obesity?;
- Can hiring records prove an employer guilty of sex discrimination?; and
- I am about to quit my job, but should I?

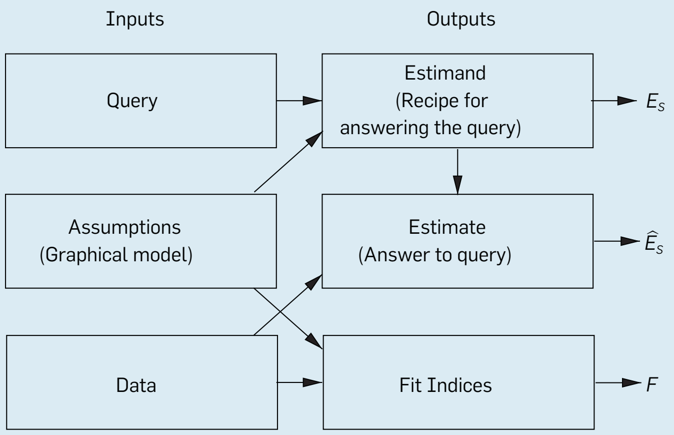
The common feature of these questions concerns cause-and-effect relationships. We recognize them through such words as "preventing," "cause," "attributed to," "discrimination," and "should I." Such words are common in everyday language, and modern society constantly demands answers to such questions. Yet, until very recently, science gave us no means even to articulate them, let alone answer them. Unlike the rules of geometry, mechanics, optics, or probabilities, the rules of cause and effect have been denied the benefits of mathematical analysis.

To appreciate the extent of this denial readers would likely be stunned to learn that only a few decades ago scientists were unable to write down a mathematical equation for the obvious fact that "Mud does not cause rain." Even today, only the top echelon of the scientific community can write such an equation and formally distinguish "mud causes rain" from "rain causes mud."

Figure 1. The causal hierarchy. Questions at level 1 can be answered only if information from level i or higher is available.

Level (Symbol)	Typical Activity	Typical Questions	Examples
1. Association $P(y\|x)$	Seeing	What is? How would seeing *X* change my belief inY?	What does a symptom tell me about a disease? What does a survey tell us about the election results?
2. Intervention $P(y\|do(x), z)$	Doing, Intervening	What if? What if I do *X*?	What if I take aspirin, will my headache be cured? What if we ban cigarettes?
3. Counterfactuals $P(y_x\|x', y')$	Imagining, Retrospection	Why? Was it *X* that caused *Y*? What if I had acted differently?	Was it the aspirin that stopped my headache? Would Kennedy be alive had Oswald not shot him? What if I had not been smoking the past two years?

Figure 2. How the SCM "inference engine" combines data with a causal model (or assumptions) to produce answers to queries of interest.



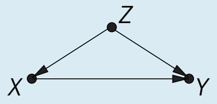
These impediments have changed dramatically in the past three decades; for example, a mathematical language has been developed for managing causes and effects, accompanied by a set of tools that turn causal analysis into a mathematical game, like solving algebraic equations or finding proofs in high-school geometry. These tools permit scientists to express causal questions formally, codify their existing knowledge in both diagrammatic and algebraic forms, and then leverage data to estimate the answers. Moreover, the theory warns them when the state of existing knowledge or the available data is insufficient to answer their questions and then suggests additional sources of knowledge or data to make the questions answerable.

The development of the tools has had a transformative impact on all data-intensive sciences, especially social science and epidemiology, in which causal diagrams have become a second language.[14,34] In these disciplines, causal diagrams have helped scientists extract causal relations from associations and deconstruct paradoxes that have baffled researchers for decades.[23,25]

I call the mathematical framework that led to this transformation "structural causal models" (SCM), which consists of three parts: graphical models, structural equations, and counterfactual and interventional logic. Graphical models serve as a language for representing what agents know about the world. Counterfactuals help them articulate what they wish to know. And structural equations serve to tie the two together in a solid semantics.

Figure 2 illustrates the operation of SCM in the form of an inference engine. The engine accepts three inputs—Assumptions, Queries, and Data—and produces three outputs—Estimand, Estimate, and Fit indices.

Figure 3. Graphical model depicting causal assumptions about three variables; the task is to estimate the causal effect of *X* on *Y* from non-experimental data on {*X*, *Y*, *Z*}.



The Estimand (E_S) is a mathematical formula that, based on the Assumptions, provides a recipe for answering the Query from any hypothetical data, whenever it is available. After receiving the data, the engine uses the Estimand to produce an actual Estimate ($\hat{E}_S$) for the answer, along with statistical estimates of the confidence in that answer, reflecting the limited size of the dataset, as well as possible measurement errors or missing data. Finally, the engine produces a list of "fit indices" that measure how compatible the data is with the Assumptions conveyed by the model.

To exemplify these operations, assume our Query stands for the causal effect of X (taking a drug) on Y (recovery), written as $Q = P(Y|do(X))$. Let the modeling assumptions be encoded (see Figure 3), where Z is a third variable (say, Gender) affecting both X and Y. Finally, let the data be sampled at random from a joint distribution $P(X, Y, Z)$. The Estimand (E_S) derived by the engine (automatically using Tool 2, as discussed in the next section) will be the formula $E_S = \Sigma z\, P(Y|X, Z)P(Z)$, which defines a procedure of estimation. It calls for estimating the gender-specific conditional distributions $P(Y|X, Z)$ for males and females, weighing them by the probability $P(Z)$ of membership in each gender, then taking the average. Note the Estimand E_S defines a property of $P(X,Y, Z)$ that, if properly estimated, would provide a correct answer to our Query. The answer itself, the Estimate $\hat{E}_S$, can be produced through any number of techniques that produce a consistent estimate of ES from finite samples of $P(X,Y, Z)$. For example, the sample average (of Y) over all cases satisfying the specified X and Z conditions would be a consistent estimate. But more-efficient estimation techniques can be devised to overcome data sparsity.[28] This task of estimating statistical relationships from sparse data is where deep learning techniques excel, and where they are often employed.[33]

Finally, the Fit Index for our example in Figure 3 will be NULL; that is, after examining the structure of the graph in Figure 3, the engine should conclude (using Tool 1, as discussed in the next section) that the assumptions encoded lack testable implications. Therefore, the veracity of the resultant estimate must lean entirely on the assumptions encoded in the arrows of Figure 3, so neither refutation nor corroboration can be obtained from the data.[c]

The same procedure applies to more sophisticated queries, as in, say, the counterfactual query $Q = P(y_x\, |x',y')$ discussed earlier. We may also permit some of the data to arrive from controlled experiments that would take the form $P(V|do(W))$ in case W is the controlled variable. The role of the Estimand would remain that of converting the Query into the syntactic form involving the available data and then guiding the choice of the estimation technique to ensure unbiased estimates. The conversion task is not always feasible, in which case the Query is declared "non-identifiable," and the engine should exit with FAILURE. Fortunately, efficient and complete algorithms have been developed to decide identifiability and produce Estimands for a variety of counterfactual queries and a variety of data types.[3,30,32]

I next provide a bird's-eye view of seven tasks accomplished through the SCM framework and the tools used in each task and discuss the unique contribution each tool brings to the art of automated reasoning.

Tool 1. Encoding causal assumptions: Transparency and testability. The task of encoding assumptions in a compact and usable form is not a trivial matter once an analyst takes seriously the requirement of transparency and testability.[d] Transparency enables analysts to discern whether the assumptions encoded are plausible (on scientific grounds) or whether additional assumptions are warranted. Testability permits us (whether analyst or machine) to determine whether the assumptions encoded are compatible with the available data and, if not, identify those that need repair.

Advances in graphical models have made compact encoding feasible. Their transparency stems naturally from the fact that all assumptions are encoded qualitatively in graphical form, mirroring the way researchers perceive cause-effect relationships in the domain; judgments of counterfactual or statistical dependencies are not required, since such dependencies can be read off the structure of the graph.[18] Testability is facilitated through a graphical criterion called *d*-separation that provides the fundamental connection between causes and probabilities. It tells us, for any given pattern of paths in the model, what pattern of dependencies we should expect to find in the data.[15]

Tool 2. *Do*-calculus and the control of confounding. Confounding, or the presence of unobserved causes of two or more variables, long considered *the* major obstacle to drawing causal inference from data, has been demystified and "deconfounded" through a graphical criterion called "backdoor." In particular, the task of selecting an appropriate set of covariates to control for confounding has been reduced to a simple "roadblocks" puzzle manageable through a simple algorithm.[16]

For models where the backdoor criterion does not hold, a symbolic engine is available, called "*do*-calculus," that predicts the effect of policy interventions whenever feasible and exits with failure whenever predictions cannot be ascertained on the basis of the specified assumptions.[3,17,30,32]

Tool 3. The algorithmitization of counterfactuals. Counterfactual analysis deals with behavior of specific individuals identified by a distinct set of characteristics. For example, given that Joe's salary is $Y = y$, and that he went $X = x$ years to college, what would Joe's salary be had he had one more year of education?

One of the crowning achievements of contemporary work on causality has been to formalize counterfactual reasoning within the graphical representation, the very representation researchers use to encode scientific knowledge. Every structural equation model determines the "truth value" of every counterfactual sentence. Therefore, an algorithm can determine if the

[c] The assumptions encoded in Figure 3 are conveyed by its missing arrows. For example, Y does not influence X or Z, X does not influence Z, and, most important, Z is the only variable affecting both X and Y. That these assumptions lack testable implications can be concluded directly from the fact that the graph is complete; that is, there exists an edge connecting every pair of nodes.

[d] Economists, for example, having chosen algebraic over graphical representations, are deprived of elementary testability-detecting features.[21]

probability of the sentence is estimable from experimental or observational studies, or a combination thereof.[1,18,30]

Of special interest in causal discourse are counterfactual questions concerning "causes of effects," as opposed to "effects of causes." For example, how likely it is that Joe's swimming exercise was a necessary (or sufficient) cause of Joe's death.[7,20]

Tool 4. Mediation analysis and the assessment of direct and indirect effects. Mediation analysis concerns the mechanisms that transmit changes from a cause to its effects. The identification of such an intermediate mechanism is essential for generating explanations, and counterfactual analysis must be invoked to facilitate this identification. The logic of counterfactuals and their graphical representation have spawned algorithms for estimating direct and indirect effects from data or experiments.[19,27,34] A typical query computable through these algorithms is: What fraction of the effect of X on Y is mediated by variable Z?

Tool 5. Adaptability, external validity, and sample selection bias. The validity of every experimental study is challenged by disparities between the experimental and the intended implementational setups. A machine trained in one environment cannot be expected to perform well when environmental conditions change, unless the changes are localized and identified. This problem, and its various manifestations, are well-recognized by AI researchers, and enterprises (such as "domain adaptation," "transfer learning," "life-long learning," and "explainable AI")[4] are just some of the subtasks identified by researchers and funding agencies in an attempt to alleviate the general problem of robustness. Unfortunately, the problem of robustness, in its broadest form, requires a causal model of the environment and cannot be properly addressed at the level of Association. Associations alone cannot identify the mechanisms responsible for the changes that occurred,[22] the reason being that surface changes in observed associations do not uniquely identify the underlying mechanism responsible for the change. The *do*-calculus discussed earlier now offers a complete methodology for overcoming bias due to environmental changes. It can be used for both for readjusting learned policies to circumvent environmental changes and for controlling disparities between nonrepresentative samples and a target population.[3] It can also be used in the context of reinforcement learning to evaluate policies that invoke new actions, beyond those used in training.[35]

Unlike the rules of geometry, mechanics, optics, or probabilities, the rules of cause and effect have been denied the benefits of mathematical analysis.

Tool 6. Recovering from missing data. Problems due to missing data plague every branch of experimental science. Respondents do not answer every item on a questionnaire, sensors malfunction as weather conditions worsen, and patients often drop from a clinical study for unknown reasons. The rich literature on this problem is wedded to a model-free paradigm of associational analysis and, accordingly, is severely limited to situations where "missingness" occurs at random; that is, independent of values taken by other variables in the model.[6] Using causal models of the missingness process we can now formalize the conditions under which causal and probabilistic relationships can be recovered from incomplete data and, whenever the conditions are satisfied, produce a consistent estimate of the desired relationship.[12,13]

Tool 7. Causal discovery. The *d*-separation criterion described earlier enables machines to detect and enumerate the testable implications of a given causal model. This opens the possibility of inferring, with mild assumptions, the set of models that are compatible with the data and to represent this set compactly. Systematic searches have been developed that, in certain circumstances, can prune the set of compatible models significantly to the point where causal queries can be estimated directly from that set.[9,18,24,31]

Alternatively, Shimizu et al.[29] proposed a method for discovering causal directionality based on functional decomposition.[24] The idea is that in a linear model $X \rightarrow Y$ with non-Gaussian noise, $P(y)$ is a convolution of two non-Gaussian distributions and would be, figuratively speaking, "more Gaussian" than $P(x)$. The relation of "more Gaussian than" can be given precise numerical measure and used to infer directionality of certain arrows.

Tian and Pearl[32] developed yet another method of causal discovery based on the detection of "shocks," or spontaneous local changes in the environment that act like "nature's interventions," and unveil causal directionality toward the consequences of those shocks.

Conclusion

I have argued that causal reasoning is an indispensable component of human thought that should be formalized and algorithmitized toward achieving human-level machine intelligence. I have explicated some of the impediments toward that goal in the form of a three-level hierarchy and showed that inference to level 2 and level 3 requires a causal model of one's environment. I have described seven cognitive tasks that require tools from these two levels of inference and demonstrated how they can be accomplished in the SCM framework.

It is important for researchers to note that the models used in accomplishing these tasks are structural (or conceptual) and require no commitment to a particular form of the distributions involved. On the other hand, the validity of all inferences depends critically on the veracity of the assumed structure. If the true structure differs from the one assumed, and the data fits both equally well, substantial errors may result that can sometimes be assessed through a sensitivity analysis.

It is also important for them to keep in mind that the theoretical limitations of model-free machine learning do not apply to tasks of prediction, diagnosis, and recognition, where interventions and counterfactuals assume a secondary role.

However, the model-assisted methods by which these limitations are circumvented can nevertheless be transported to other machine learning tasks where problems of opacity, robustness, explainability, and missing data are critical. Moreover, given the transformative impact that causal modeling has had on the social and health sciences,[14,25,34] it is only natural to expect a similar transformation to sweep through machine learning technology once it is guided by provisional models of reality. I expect this symbiosis to yield systems that communicate with users in their native language of cause and effect and, leveraging this capability, to become the dominant paradigm of next-generation AI.

Acknowledgments

This research was supported in part by grants from the Defense Advanced Research Projects Agency [#W911NF-16-057], National Science Foundation [#IIS-1302448, #IIS-1527490, and #IIS-1704932], and Office of Naval Research [#N00014-17-S-B001]. The article benefited substantially from comments by the anonymous reviewers and conversations with Adnan Darwiche of the University of California, Los Angeles.

References

1. Balke, A. and Pearl, J. Probabilistic evaluation of counterfactual queries. In *Proceedings of the 12th National Conference on Artificial Intelligence* (Seattle, WA, July 31–Aug. 4). MIT Press, Menlo Park, CA, 1994, 230–237.
2. Bareinboim, E. and Pearl, J. Causal inference by surrogate experiments: z-identifiability. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, N. de Freitas and K. Murphy, Eds. (Catalina Island, CA, Aug. 14–18). AUAI Press, Corvallis, OR, 2012, 113–120.
3. Bareinboim, E. and Pearl, J. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences 113*, 27 (2016), 7345–7352.
4. Chen, Z. and Liu, B. *Lifelong Machine Learning.* Morgan and Claypool Publishers, San Rafael, CA, 2016.
5. Darwiche, A. *Human-Level Intelligence or Animal-Like Abilities? Technical Report.* Department of Computer Science, University of California, Los Angeles, CA, 2017; https://arxiv.org/pdf/1707.04327.pdf
6. Graham, J. *Missing Data: Analysis and Design (Statistics for Social and Behavioral Sciences).* Springer, 2012.
7. Halpern, J.H. and Pearl, J. Causes and explanations: A structural-model approach: Part I: Causes. *British Journal of Philosophy of Science 56* (2005), 843–887.
8. Hutson, M. AI researchers allege that machine learning is alchemy. *Science* (May 3, 2018); https://www.sciencemag.org/news/2018/05/ai-researchers-allege-machine-learning-alchemy
9. Jaber, A., Zhang, J.J., and Bareinboim, E. Causal identification under Markov equivalence. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, A. Globerson and R. Silva, Eds. (Monterey, CA, Aug. 6–10). AUAI Press, Corvallis, OR, 2018, 978–987.
10. Lake, B.M., Salakhutdinov, R., and Tenenbaum, J.B. Human-level concept learning through probabilistic program induction. *Science 350*, 6266 (Dec. 2015), 1332–1338.
11. Marcus, G. *Deep Learning: A Critical Appraisal. Technical Report.* Departments of Psychology and Neural Science, New York University, New York, 2018; https://arxiv.org/pdf/1801.00631.pdf
12. Mohan, K. and Pearl, J. *Graphical Models for Processing Missing Data. Technical Report R-473.* Department of Computer Science, University of California, Los Angeles, CA, 2018; forthcoming, *Journal of American Statistical Association*; http://ftp.cs.ucla.edu/pub/stat_ser/r473.pdf
13. Mohan, K., Pearl, J., and Tian, J. Graphical models for inference with missing data. In *Advances in Neural Information Processing Systems 26*, C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, Eds. Curran Associates, Inc., Red Hook, NY, 2013, 1277–1285; http://papers.nips.cc/paper/4899-graphical-models-for-inference-with-missing-data.pdf
14. Morgan, S.L. and Winship, C. *Counterfactuals and Causal Inference: Methods and Principles for Social Research (Analytical Methods for Social Research), Second Edition.* Cambridge University Press, New York, 2015.
15. Pearl, J. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Mateo, CA, 1988.
16. Pearl, J. Comment: Graphical models, causality, and intervention. *Statistical Science 8*, 3 (1993), 266–269.
17. Pearl, J. Causal diagrams for empirical research. *Biometrika 82*, 4 (Dec. 1995), 669–710.
18. Pearl, J. *Causality: Models, Reasoning, and Inference.* Cambridge University Press, New York, 2000; *Second Edition*, 2009.
19. Pearl, J. Direct and indirect effects. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence* (Seattle, WA, Aug. 2–5). Morgan Kaufmann, San Francisco, CA, 2001, 411–420.
20. Pearl, J. Causes of effects and effects of causes. *Journal of Sociological Methods and Research 44*, 1 (2015a), 149–164.
21. Pearl, J. Trygve Haavelmo and the emergence of causal calculus. *Econometric Theory 31*, 1 (2015b), 152–179; special issue on Haavelmo centennial
22. Pearl, J. and Bareinboim, E. External validity: From *do*-calculus to transportability across populations. *Statistical Science 29*, 4 (2014), 579–595.
23. Pearl, J. and Mackenzie, D. *The Book of Why: The New Science of Cause and Effect.* Basic Books, New York, 2018.
24. Peters, J., Janzing, D. and Schölkopf, B. *Elements of Causal Inference: Foundations and Learning Algorithms.* MIT Press, Cambridge, MA, 2017.
25. Porta, M. The deconstruction of paradoxes in epidemiology. OUPblog, Oct. 17, 2014; https://blog.oup.com/2014/10/deconstruction-paradoxes-sociology-epidemiology/
26. Ribeiro, M.T., Singh, S., and Guestrin, C. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, CA, Aug. 13–17). ACM Press, New York, 2016, 1135–1144.
27. Robins, J.M. and Greenland, S. Identifiability and exchangeability for direct and indirect effects. *Epidemiology 3*, 2 (Mar. 1992), 143–155.
28. Rosenbaum, P. and Rubin, D. The central role of propensity score in observational studies for causal effects. *Biometrika 70*, 1 (Apr. 1983), 41–55.
29. Shimizu, S., Hoyer, P.O., Hyvärinen, A., and Kerminen, A.J. A linear non-Gaussian acyclic model for causal discovery. *Journal of the Machine Learning Research 7* (Oct. 2006), 2003–2030.
30. Shpitser, I. and Pearl, J. Complete identification methods for the causal hierarchy. *Journal of Machine Learning Research 9* (2008), 1941–1979.
31. Spirtes, P., Glymour, C.N., and Scheines, R. *Causation, Prediction, and Search, Second Edition.* MIT Press, Cambridge, MA, 2000.
32. Tian, J. and Pearl, J. A general identification condition for causal effects. In *Proceedings of the 18th National Conference on Artificial Intelligence* (Edmonton, AB, Canada, July 28–Aug. 1). AAAI Press/MIT Press, Menlo Park, CA, 2002, 567–573.
33. van der Laan, M.J. and Rose, S. *Targeted Learning: Causal Inference for Observational and Experimental Data.* Springer, New York, 2011.
34. VanderWeele, T.J. *Explanation in Causal Inference: Methods for Mediation and Interaction.* Oxford University Press, New York, 2015.
35. Zhang, J. and Bareinboim, E. Transfer learning in multi-armed bandits: A causal approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (Melbourne, Australia, Aug. 19–25). AAAI Press, Menlo Park, CA, 2017, 1340–1346.

Judea Pearl (judea@cs.ucla.edu) is a professor of computer science and statistics and director of the Cognitive Systems Laboratory at the University of California, Los Angeles, USA.

Watch the author discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/the-seven-tools-of-causal-inference

DOI:10.1145/3241979

Metamorphic testing can test untestable software, detecting fatal errors in autonomous vehicles' onboard computer systems.

BY ZHI QUAN ZHOU AND LIQUN SUN

Metamorphic Testing of Driverless Cars

ON MARCH 18, 2018, Elaine Herzberg became the first pedestrian in the world to be killed by an autonomous vehicle after being hit by a self-driving Uber SUV in Tempe, AZ, at about 10 P.M. Video released by the local police department showed the self-driving Volvo XC90 did not appear to see Herzberg, as it did not slow down or alter course, even though she was visible in front of the vehicle prior to impact. Subsequently, automotive engineering experts raised questions about Uber's LiDAR technology.[12] LiDAR, or "light detection and ranging," uses pulsed laser light to enable a self-driving car to see its surroundings hundreds of feet away.

Velodyne, the supplier of the Uber vehicle's LiDAR technology, said, "Our LiDAR is capable of clearly imaging Elaine and her bicycle in this situation. However, our LiDAR does not make the decision to put on the brakes or get out of her way" ... "We know absolutely nothing about the engineering of their [Uber's] part ... It is a proprietary secret, and all of our customers keep this part to themselves"[15] ... and "Our LiDAR can see perfectly well in the dark, as well as

» key insights

- Many software systems (such as AI systems and those that control self-driving vehicles) are difficult to test using conventional approaches and are known as "untestable software."
- Metamorphic testing can test untestable software in a very cost-effective way, using a perspective not previously used by conventional approaches.
- We detected fatal software faults in the LiDAR obstacle-perception module of self-driving cars and reported the alarming results eight days before Uber's deadly crash in Tempe, AZ, in March 2018.

it sees in daylight, producing millions of points of information. However, it is up to the rest of the system to interpret and use the data to make decisions. We do not know how the Uber system of decision making works."[11]

Question Concerning Every Human Life

Regardless of investigation outcomes, this Uber fatal accident raised a serious question about the perception capability of self-driving cars: Are there situations where a driverless car's onboard computer system could incorrectly "interpret and use" the data sent from a sensor (such as a LiDAR sensor), making the car unable to detect a pedestrian or obstacle in the roadway? This question is not specific to Uber cars but is general enough to cover all types of autonomous vehicles, and the answer concerns every human life. Unfortunately, our conclusion is affirmative. Even though we could not access the Uber system, we have managed to test Baidu Apollo, a well-known real-world self-driving software system controlling many autonomous vehicles on the road today (http://apollo.auto). Using a novel metamorphic testing method, we have detected critical software errors that could cause the Apollo perception module to misinterpret the point cloud data sent from the LiDAR sensor, making some pedestrians and obstacles undetectable. The Apollo system uses Velodyne's HDL64E LiDAR sensor,[1] exactly the same type of LiDAR involved in the Uber accident.[16]

We reported this issue to the Baidu Apollo self-driving car team on March 10, 2018, MST (UTC -7), eight days before the Uber accident. Our bug report was logged as issue #3341 (https://github.com/ApolloAuto/apollo/issues/3341). We did not receive a response from Baidu until 10:25 P.M., March 19, 2018, MST—24 hours after the Uber accident. In its reply, the Apollo perception team confirmed the error. Before presenting further details of our findings, we first discuss the challenges of testing complex computer systems, with a focus on software testing for autonomous vehicles.

Testing Challenge

Testing is a major approach to software quality assurance. Deploying inadequately tested software can have serious consequences.[22] Software testing is, however, fundamentally challenged by the "oracle problem." An oracle is a mechanism testers use to determine whether the outcomes of test-case executions are correct.[2,24] Most software-testing techniques assume an oracle exists. However, this assumption does not always hold when testing complex applications. This is thus the oracle problem, a situation where an oracle is unavailable or too expensive to be applied. For example, when a software engineer is testing a compiler, determining the equivalence between the source code and the compiler-generated object program is difficult. When testing a Web search engine, the tester finds it very difficult to assess the completeness of the search results.

To achieve a high standard of testing, the tester needs to generate, execute, and verify a large number of tests. These tasks can hardly be done without test automation. For testing self-driving vehicles, constructing a fully automated test oracle is especially difficult. Although in some situations the human tester can serve as an oracle to verify the vehicle's behavior, manual monitoring is expensive and error-prone. In the Uber accident, for example, the safety driver performed no safety monitoring, and because "humans monitoring an automated system are likely to become bored and disengaged," such testing is "particularly dangerous."[12]

The oracle problem is also reflected in the difficulty of creating detailed system specifications against which the autonomous car's behavior can be checked, as it essentially involves recreating the logic of a human driver's decision making.[21] Even for highly qualified human testers with full system specifications, it can still be difficult or even impossible to determine the correctness of every behavior of an autonomous vehicle. For example, in a complex road network, it is difficult for the tester to decide whether the driving route selected by the autonomous car is optimal.[3] Likewise, it is not easy to verify whether the software system has correctly interpreted the huge amount of point-cloud data sent from a LiDAR sensor, normally at a rate of more than one million data points per second.

"Negative testing" is even more challenging. While positive testing focuses on ensuring a program does what it is supposed to do for normal input, negative testing serves to ensure the program does not do what it is *not* supposed to do when the input is unexpected, normally involving random factors or events. Resource constraints and deadline pressures often result in development organizations skipping negative testing, potentially allowing safety and security issues to persist into the released software.[5,22]

In the context of negative software testing for autonomous vehicles (if attempted by the development organization), how can the tester identify the conditions under which the vehicle could potentially do something wrong, as in, say, unintentionally striking a pedestrian? To a certain degree, tools called "fuzzers" could help perform this kind of negative software testing. During "fuzzing," or "fuzz testing," the fuzzer generates a random or semi-random input and feeds it into the system under test, hoping to crash the system or cause it to misbehave.[22] However, the oracle problem makes verification of the fuzz test results (outputs for millions of random inputs) extremely difficult, if not impossible.[5] In fuzzing, the tester thus looks only for software crashes (such as aborts and hangs). This limitation means not only huge numbers of test cases might need to be run before a crash but also that logic errors, which do not crash the system but instead produce incorrect output, cannot be detected.[5] For example, fuzzing cannot detect the error when a calculator returns "1 + 1 = 3." Neither can simple fuzzing detect misinterpretation of LiDAR data.

Metamorphic Testing

Metamorphic testing (MT)[6] is a property-based software-testing technique that can effectively address two fundamental problems in software testing: the oracle problem and the automated test-case-generation problem. The main difference between MT and other testing techniques is that the former does not focus on the verification of each individual output of the software under test and can thus be performed in the absence of an oracle. MT checks the relations among the inputs and outputs of multiple executions of the software. Such relations are called "metamorphic relations" (MRs) and are necessary properties of the intended program's functionality. If, for certain test cases, an MR is violated, then the software must be faulty. Consider, for example, the testing of a search engine. Suppose the tester entered a search criterion C_1 and the search engine returned 50,000 results. It may not be easy to verify the accuracy and completeness of these 50,000 results. Nevertheless, an MR can be identified as follows: The search results for C_1 must include those for C_1 `AND` C_2, where C_2 can be any additional condition (such as a string or a filter). If the actual search results violate this relation, the search engine must be at fault. Here, the search criterion "C_1 `AND` C_2" is a new test case that can be constructed automatically based on the source test case "C_1" (whereas C_2 could be generated automatically and randomly) and the satisfaction of the MR can also be verified automatically through a testing program.

A growing body of research from both industry and academia has examined the MT concept and proved it highly effective.[4,7–9,13,18–20] The increasing interest in MT is not only due to it being able to address the oracle problem and automate test generation but also the perspective of MT has seldom been used in previous testing strategies and, as a result, has detected a large number of previously unknown faults in many mature systems (such as the GCC and LLVM compilers),[10,17] the Web search engines Google and Bing,[25] and code obfuscators.[5]

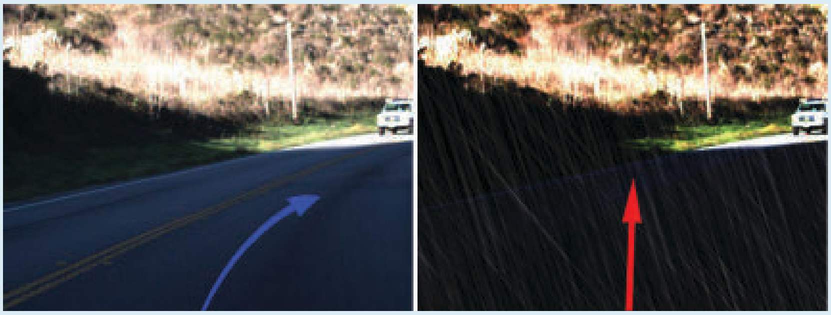
MT for Testing Autonomous Machinery

Several research groups have begun to apply MT to alleviate the difficulties in testing autonomous systems, yielding encouraging results.

For example, researchers from the Fraunhofer Center for Experimental Software Engineering in College Park, MD, developed a simulated environment in which the control software of autonomous drones was tested using MT.[14] The MRs made use of geometric transformations (such as rotation and translation) in combination with different formations of obstacles in the flying scenarios of the drone. The Fraunhofer researchers looked for behavioral differences of the drone when it was flying under these different (supposedly equivalent) scenarios. Their MRs required the drone should have consistent behavior, while finding that in some situations the drone behaved inconsistently, revealing multiple software defects. For example, one of the bugs was in the sense-and-avoid algorithm, making the algorithm sensitive to certain numerical values and hence misbehavior under certain conditions, causing the drone to crash. The researchers detected another bug after running hundreds of tests using different rotations of the environment: The drone had landing problems in some situations. This was because the researchers rotated all the objects in the environment, but not the sun, unexpectedly causing a shadow to fall on the landing pad in some orientations, revealing issues in the drone's vision system. The researchers solved this with a more robust vision sensor that was less sensitive to lighting changes.

Researchers from the University of Virginia and from Columbia University tested three different deep neural network (DNN) models for autonomous driving.[21] The inputs to the models were pictures from a camera, and the outputs were steering angles. To verify the correctness of the outputs, the researchers used a set of MRs based on

Figure 1. Input pictures used in a metamorphic test revealing inconsistent and erroneous behavior of a DNN (https://deeplearningtest.github.io/deepTest).[21]



(a) original

(b) with added rain

Figure 2. MT detected a real-life bug in Google Maps;[3,20] the origin and destination of the route were almost at the same point, but Google Maps generated an "optimal" route of 4.9 miles.

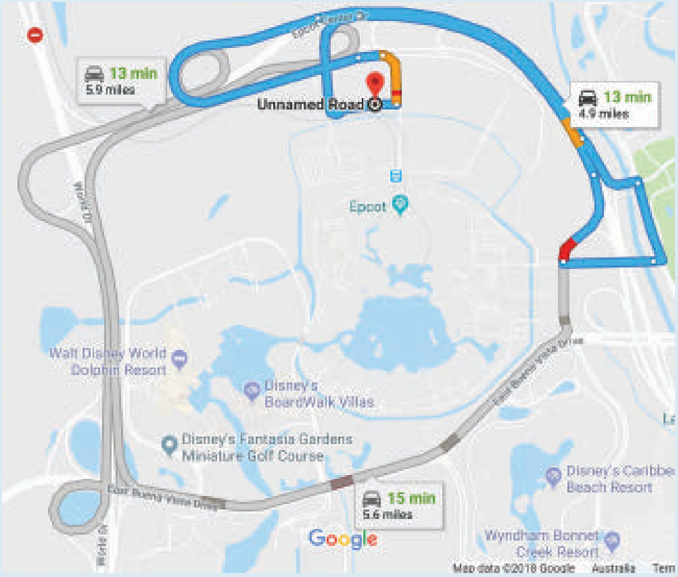
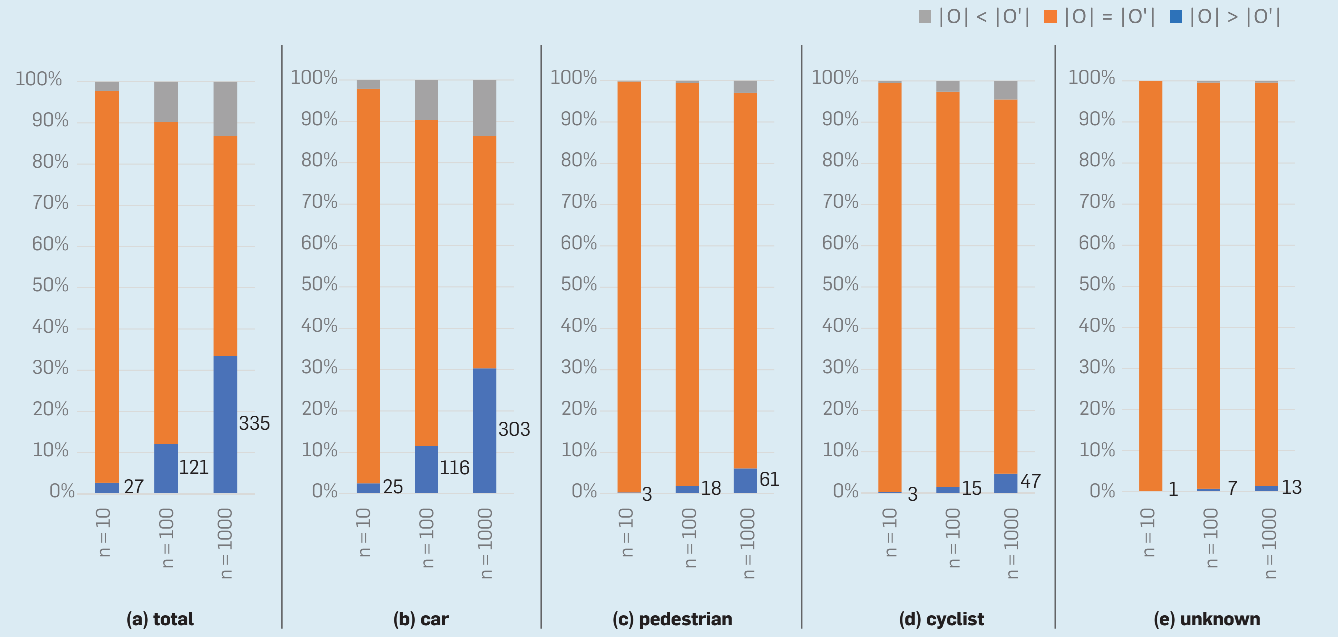

Epcot Center Dr
13 min
5.9 miles
Unnamed Road
13 min
4.9 miles
World Dr
Epcot
Walt Disney World Dolphin Resort
Disney's BoardWalk Villas
East Buena Vista Drive
Disney's Fantasia Gardens Miniature Golf Course
Disney's Caribbe Beach Resort
East Buena Vista Drive
15 min
5.6 miles
Google
Wyndham Bonnet Creek Resort
World Dr
Map data ©2018 Google Australia Term

Figure 3. Results of experiments by category, visualized as 100% stacked column charts.

Each vertical column represents the comparisons of 1,000 pairs of results, where the blue subsection implies MR_1 violations.
Each blue subsection is labeled with the actual number of $|O| > |O'|$ cases (out of the 1,000 pairs).



(a) total (b) car (c) pedestrian (d) cyclist (e) unknown

image transformations. The MRs said the car should behave similarly under variations of the same input (such as the same scene under different lighting conditions). Using these MRs, they generated realistic synthetic images based on seed images. These synthetic images mimic real-world phenomena (such as camera lens distortions and different weather conditions). Using MT, together with a notion of neuron coverage (the number of neurons activated), the researchers found a large number of "corner case" inputs leading to erroneous behavior in three DNN models. Figure 1 is an example, whereby the original trajectory (the blue arrow) and the second trajectory (the red arrow) are inconsistent, revealing dangerous erroneous behavior of the DNN model under test.

We recently applied MT to test Google Maps services that can be used to navigate autonomous cars,[3] identifying a set of MRs for the navigation system. For example, one of the MRs was "that a restrictive condition such as avoiding tolls should not result in a more optimal route." With these MRs, we detected a large number of real-life bugs in Google Maps, one shown in Figure 2; the origin and destination points were almost at the same location, but Google Maps returned a route of 4.9 miles, which was obviously unacceptable.

LiDAR-Data-Interpretation Errors

The scope of the study conducted by the researchers from the University of Virginia and from Columbia University[21] was limited to DNN models. A DNN model is only part of the perception module of a self-driving car's software system. Moreover, while a DNN can take input from different sensors (such as a camera and LiDAR), they studied only the ordinary two-dimensional picture input from a camera, and the output considered was the steering angle calculated by the DNN model based on the input picture. The software tested was not real-life systems controlling autonomous cars but rather deep learning models that "won top positions in the Udacity self-driving challenge."

Unlike their work, this article reports our testing of the real-life Apollo system, which is the onboard software in Baidu's self-driving vehicles. Baidu also claims users can directly use the software to build their own autonomous cars (http://apollo.auto/cooperation/detail_en_01.html).

Software under test. More specifically, we tested Apollo's perception module (http://apollo.auto/platform/perception.html), which has two key components: "three-dimensional obstacle perception" and "traffic-light perception." We tested the three-dimensional obstacle perception component, which itself consisted of three subsystems: "LiDAR obstacle perception," "RADAR obstacle perception," and "obstacle results fusion." Although our testing method is applicable to all three subsystems, we tested

Test results; for each value of *n*, we compared 1,000 pairs of results.

number of added points (n)	|O| > |O'|	|O| = |O'|	|O| < |O'|
10	27	951	22
100	121	781	98
1,000	335	533	132

only the first, LiDAR obstacle perception (or LOP), which takes the three-dimensional point cloud data as input, as generated by Velodyne's HDL64E LiDAR sensor.

LOP resolves the raw point-cloud data using the following pipeline, as excerpted from the Apollo website (https://github.com/ApolloAuto/apollo/blob/master/docs/specs/3d_obstacle_perception.md):

HDMap region of interest filter (tested in our experiments). The region of interest (ROI) specifies the drivable area, including road surfaces and junctions that are retrieved from a high-resolution (HD) map. The HDMap ROI filter processes LiDAR points that are outside the ROI, removing background objects (such as buildings and trees along the road). What remains is the point cloud in the ROI for subsequent processing;

Convolutional neural networks segmentation (tested in our experiments). After identifying the surrounding environment using the HDMap ROI filter, the Apollo software obtains the filtered point cloud that includes only the points inside the ROI—the drivable road and junction areas. Most of the background obstacles (such as buildings and trees along the road) have been removed, and the point cloud inside the ROI is fed into the "segmentation" module. This process detects and segments out foreground obstacles (such as cars, trucks, bicycles, and pedestrians). Apollo uses a deep convolutional neural network (CNN) for accurate obstacle detection and segmentation. The output of this process is a set of objects corresponding to obstacles in the ROI;

MinBox builder (tested in our experiments). This object builder component establishes a bounding box for the detected obstacles;

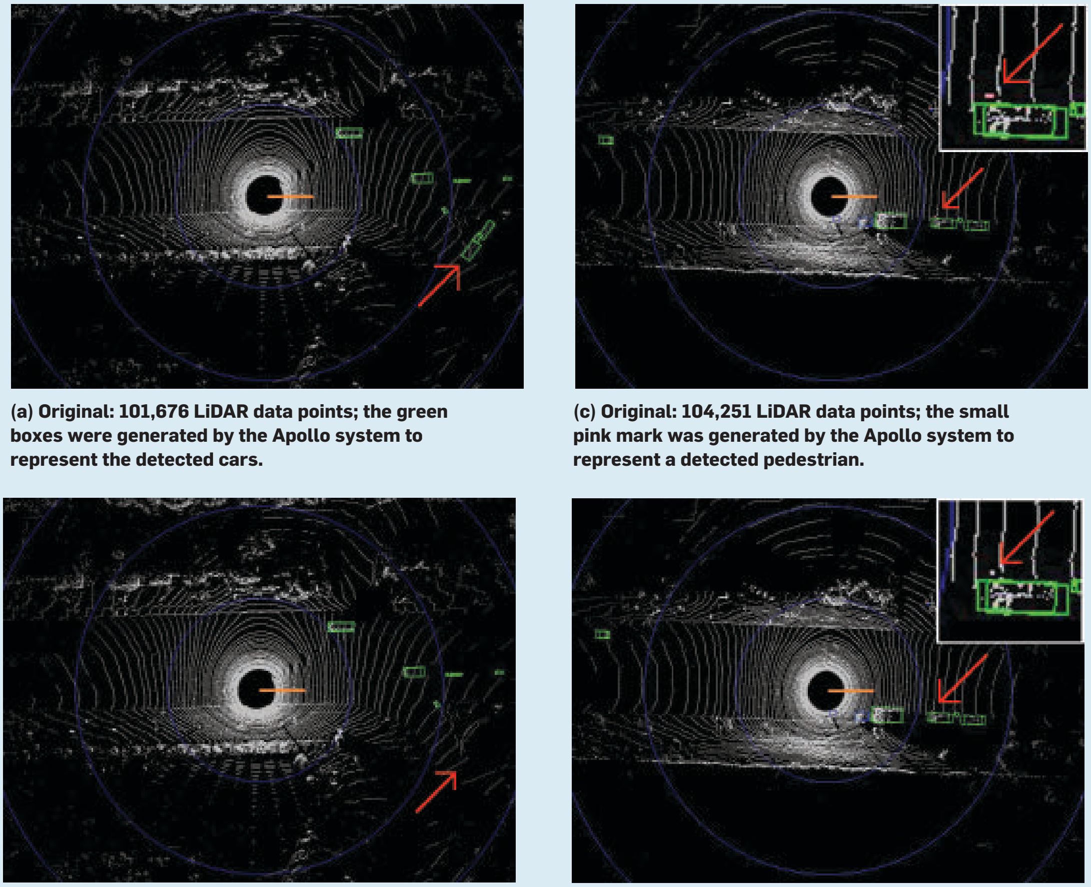
HM object tracker (not tested in our experiments). This tracker is designed to track obstacles detected in the segmentation step; and

Sequential type fusion (not tested in our experiments). To smooth the obstacle type and reduce the type switch over the entire trajectory, Apollo uses a sequential type fusion algorithm.

Our software-testing experiments involved the first, second, and third but not the fourth and fifth features, because the first three are the most critical and fundamental.

Our testing method: MT in combination with fuzzing. Based on the Baidu specification of the HDMap ROI filter, we identified the following meta-

Figure 4. MT detected real-life fatal errors in LiDAR point-cloud data interpretation in the Apollo "perception" module: three missing cars and one missing pedestrian.



(a) Original: 101,676 LiDAR data points; the green boxes were generated by the Apollo system to represent the detected cars.

(c) Original: 104,251 LiDAR data points; the small pink mark was generated by the Apollo system to represent a detected pedestrian.

(b) After adding 1,000 random data points outside the ROI, the three cars inside the ROI could no longer be detected.

(d) After adding only 10 random data points outside the ROI, the pedestrian inside the ROI could no longer be detected.

morphic relation, whereby the software under test is the LiDAR obstacle perception (LOP) subsystem of Apollo, A and A' represent two inputs to LOP, and O and O' represent LOP's outputs for A and A', respectively.

MR_1. Let A and A' be two frames of three-dimensional point cloud data that are identical except that A' includes a small number of additional LiDAR data points randomly scattered in regions outside the ROI. Also let O and O' be the sets of obstacles identified by LOP for A and A', respectively (LOP identifies only obstacles within the ROI). The following relation must then hold: $O \subseteq O'$.

In MR_1, the additional LiDAR data points in A' could represent small particles in the air or just some noise from the sensor, whose existence is possible.[23] MR_1 says the existence of some particles, or some noise points, or their combination, in the air far from the ROI should not cause an obstacle on the roadway to become undetectable. As an extreme example, a small insect 100 meters away—outside the ROI—should not interfere with the detection of a pedestrian in front of the vehicle. This requirement is intuitively valid and agrees with the Baidu specification of its HDMap ROI filter. According to the user manual for the HDL64E LiDAR sensor, it can be mounted atop the vehicle, delivering a 360° horizontal field of view and a 26.8° vertical field of view, capturing a point cloud with a range up to 120 meters.

As an extreme example, a small insect 100 meters away—outside the ROI—should not interfere with the detection of a pedestrian in front of the vehicle.

We next describe the design of three series of experiments to test the LOP using MR_1. The Apollo Data Open Platform (http://data.apollo.auto) provides a set of "vehicle system demo data"—sensor data collected at real scenes. We downloaded the main file of this dataset, named demo-sensor-demo-apollo-1.5.bag (8.93GB). This file included point cloud data collected by Baidu engineers using the Velodyne LiDAR sensor on the morning of September 6, 2017. In each series of experiments, we first randomly extracted 1,000 frames of the point cloud data; we call each such frame a "source test case." For each source test case t, we ran the LOP software to identify its ROI and generate O, the set of detected obstacles for t. We then constructed a follow-up test case t' by randomly scattering n additional points into the three-dimensional space outside the ROI of t; we determined the value of the z coordinate of each point by choosing a random value between the minimum and maximum z-coordinate values of all points in t. Using a similar approach, we also generated a d value—the reflected intensity of the laser—for each added point. We then ran the LOP software for t', producing O', the set of detected obstacles. Finally, we compared O and O'. We conducted three series of experiments: for $n = 10$, 100, and 1,000. We thus ran the LOP software for a total of (1,000 + 1,000) × 3 = 6,000 times, processing 3,000 source test cases and 3,000 follow-up test cases.

Test results. In our experiments, for ease of implementation of MR_1, we did not check the subset relation $O \subseteq O'$ but instead compared the numbers of objects contained in O and O', denoted by $|O|$ and $|O'|$, respectively. Note that $O \subseteq O' \rightarrow |O| \leq |O'|$; hence, the condition we actually checked was less strict than MR_1. That is, if $|O| > |O'|$, then there must be something wrong, as one or more objects in O must be missing in O'.

The results of our experiments were quite surprising; the table here summarizes the overall results. The violation rates (that is, cases for $|O| > |O'|$ out of 1,000 pairs of outputs) were 2.7% (= 27 ÷ 1,000), 12.1% (=121÷ 1,000), and 33.5% (= 335 ÷ 1,000), for $n = 10$, 100, and 1,000, respectively. This means as few as 10 sheer random points scattered in the vast three-dimensional space outside the ROI could cause the driverless car to fail to detect an obstacle on the roadway, with 2.7% probability. When the number of random points increased to 1,000, the probability became as high as 33.5%. According to the HDL64E user manual, the LiDAR sensor generates more than one million data points per second, and each frame of point cloud data used in our experiments normally contained more than 100,000 data points. The random points we added to the point cloud frames were thus trivial.

The LOP software in our experiments categorized the detected obstacles into four types: detected car, pedestrian, cyclist, and unknown, as "depicted by bounding boxes in green, pink, blue and purple respectively" (http://apollo.auto/platform/perception.html). Figure 3b to Figure 3e summarize the test results of these categories, and Figure 3a shows the overall results corresponding to the Table.

Each vertical column in Figure 3 includes a subsection in blue, corresponding to MR_1 violations. They are labeled with the actual numbers of $|O| > |O'|$ cases. We observed that all these numbers were greater than 0, indicating critical errors in the perception of all four types of obstacles: car, pedestrian, cyclist, and unknown. Relatively speaking, the error rate of the "car" category was greatest, followed by "pedestrian," "cyclist," and "unknown."

Figure 4a and Figure 4b show a real-world example revealed by our test, whereby three cars inside the ROI could not be detected after we added 1,000 random points outside the ROI. Figure 4c and Figure 4d show another example, whereby a pedestrian inside the ROI (the Apollo system depicted this pedestrian with the small pink mark in Figure 4c) could not be detected after we added only 10 random points outside the ROI; as shown in Figure 4d, the small pink mark was missing. As mentioned earlier, we reported the bug to the Baidu Apollo self-driving car team on March 10, 2018. On March 19, 2018, the Apollo team confirmed the error by acknowledging "It might happen" and suggested "For cases like that, models can be fine tuned using data augmentation"; data augmentation is a technique that alleviates the problem of lack of training data in machine learning by inflating the training set through transformations of the existing data. Our failure-causing metamorphic test cases (those with the random points) could thus serve this purpose.

Conclusion

The 2018 Uber fatal crash in Tempe, AZ, revealed the inadequacy of conventional testing approaches for mission-critical autonomous systems. We have shown MT can help address this limitation and enable automatic detection of fatal errors in self-driving vehicles that operate on either conventional algorithms or deep learning models. We have introduced an innovative testing strategy that combines MT with fuzzing, reporting how we used it to detect previously unknown fatal errors in the real-life LiDAR obstacle perception system of Baidu's Apollo self-driving software.

The scope of our study was limited to LiDAR obstacle perception. Apart from LiDAR, an autonomous vehicle may also be equipped with radar. According to the Apollo website (http://data.apollo.auto), "Radar could precisely estimate the velocity of moving obstacles, while LiDAR point cloud could give a better description of object shape and position." Moreover, there can also be cameras, which are particularly useful for detecting visual features (such as the color of traffic lights). Our testing technique can be applied to radar, camera, and other types of sensor data, as well as obstacle-fusion algorithms involving multiple sensors. In future research, we plan to collaborate with industry to develop MT-based testing techniques, combined with existing verification and validation methods, to make driverless vehicles safer.

Acknowledgments

This work was supported in part by a linkage grant from the Australian Research Council, project ID: LP160101691. We would like to thank Suzhou Insight Cloud Information Technology Co., Ltd., for supporting this research.

References

1. Baidu, Inc. *Apollo Reference Hardware*, Mar. 2018; http://apollo.auto/platform/hardware.html
2. Barr, E.T., Harman, M., McMinn, P., Shahbaz, M., and Yoo, S. The oracle problem in software testing: A survey. *IEEE Transactions on Software Engineering 41*, 5 (May 2015), 507–525.
3. Brown, J., Zhou, Z.Q., and Chow, Y.-W. Metamorphic testing of navigation software: A pilot study with Google Maps. In *Proceedings of the 51st Annual Hawaii International Conference on System Sciences* (Big Island, HI, Jan. 3–6, 2018) 5687–5696; http://hdl.handle.net/10125/50602
4. Chen, T.Y., Kuo, F.-C., Liu, H., Poon, P.-L., Towey, D., Tse, T.H., and Zhou, Z.Q. Metamorphic testing: A review of challenges and opportunities. *ACM Computing Surveys 51*, 1 (Jan. 2018), 4:1–4:27.
5. Chen, T.Y., Kuo, F.-C., Ma, W., Susilo, W., Towey, D., Voas, J., and Zhou, Z.Q. Metamorphic testing for cybersecurity. *Computer 49*, 6 (June 2016), 48–55.
6. Chen, T.Y., Tse, T.H., and Zhou, Z.Q. Fault-based testing without the need of oracles. *Information and Software Technology 45*, 1 (2003), 1–9.
7. Donaldson, A.F., Evrard, H., Lascu, A., and Thomson, P. Automated testing of graphics shader compilers. *Proceedings of the ACM on Programming Languages 1* (2017), 93:1–93:29.
8. Jarman, D.C., Zhou, Z.Q., and Chen, T.Y. Metamorphic testing for Adobe data analytics software. In *Proceedings of the IEEE/ACM Second International Workshop on Metamorphic Testing,* in conjunction with the *39th International Conference on Software Engineering* (Buenos Aires, Argentina, May 22). IEEE, 2017. 21–27; https://doi.org/10.1109/MET.2017.1
9. Kanewala, U., Pullum, L.L., Segura, S., Towey, D., and Zhou, Z.Q. Message from the workshop chairs. In *Proceedings of the IEEE/ACM First International Workshop on Metamorphic Testing,* in conjunction with the *38th International Conference on Software Engineering* (Austin, TX, May 16). ACM Press, New York, 2016.
10. Le, V., Afshari, M., and Su, Z. Compiler validation via equivalence modulo inputs. In *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation* (Edinburgh, U.K., June 9–11). ACM Press, New York, 2014, 216–226.
11. Lee, D. Sensor firm Velodyne 'baffled' by Uber self-driving death. *BBC News* (Mar. 23, 2018); http://www.bbc.com/news/technology-43523286
12. Levin, S. Uber crash shows 'catastrophic failure' of self-driving technology, experts say. *The Guardian* (Mar. 23, 2018); https://www.theguardian.com/technology/2018/mar/22/self-driving-car-uber-death-woman-failure-fatal-crash-arizona
13. Lindvall, M., Ganesan, D., Árdal, R., and Wiegand, R.E. Metamorphic model-based testing applied on NASA DAT — An experience report. In *Proceedings of the 37th IEEE/ACM International Conference on Software Engineering* (Firenze, Italy, May 16-24). IEEE, 2015, 129–138.
14. Lindvall, M., Porter, A., Magnusson, G., and Schulze, C. Metamorphic model-based testing of autonomous systems. In *Proceedings of the Second IEEE/ACM International Workshop on Metamorphic Testing,* in conjunction with the *39th International Conference on Software Engineering* (Buenos Aires, Argentina, May 22). IEEE, 2017, 35–41.
15. Ohnsman, A. LiDAR maker Velodyne 'baffled' by self-driving Uber's failure to avoid pedestrian. *Forbes* (Mar. 23, 2018); https://www.forbes.com/sites/alanohnsman/2018/03/23/lidar-maker-velodyne-baffled-by-self-driving-ubers-failure-to-avoid-pedestrian
16. Posky, M. LiDAR supplier defends hardware, blames Uber for fatal crash. *The Truth About Cars* (Mar. 23, 2018); http://www.thetruthaboutcars.com/2018/03/lidar-supplier-blames-uber/
17. Regehr, J. *Finding Compiler Bugs by Removing Dead Code.* Blog, June 20, 2014; http://blog.regehr.org/archives/1161
18. Segura, S., Fraser, G., Sanchez, A.B., and Ruiz-Cortés, A. A survey on metamorphic testing. *IEEE Transactions on Software Engineering 42*, 9 (Sept. 2016), 805–824.
19. Segura, S. and Zhou, Z.Q. Metamorphic testing: Introduction and applications. ACM SIGSOFT webinar, Sept. 27, 2017; https://event.on24.com/wcc/r/1451736/8B5B5925E82FC9807CF83C84834A6F3D
20. Segura, S. and Zhou, Z.Q. Metamorphic testing 20 years later: A hands-on introduction. In *Proceedings of the 40th IEEE/ACM International Conference on Software Engineering* (Gothenburg, Sweden, May 27–June 3, 2018). ACM Press, New York, 2018.
21. Tian, Y., Pei, K., Jana, S., and Ray, B. DeepTest: Automated testing of deep neural network-driven autonomous cars. In *Proceedings of the 40th IEEE/ACM International Conference on Software Engineering* (Gothenburg, Sweden, May 27–June 3, 2018). ACM Press, New York, 2018.
22. Vassilev, A. and Celi, C. Avoiding cyberspace catastrophes through smarter testing. *Computer 47*, 10 (Oct. 2014), 102–106.
23. Velodyne, *Velodyne's HDL-64E: A High-Definition LiDAR Sensor for 3-D Applications*, White Paper, 2007; https://www.velodynelidar.com/
24. Zhou, Z.Q., Towey, D., Poon, P.-L., and Tse, T.H. Introduction to the special issue on test oracles. *Journal of Systems and Software 136* (Feb. 2018), 187; https://doi.org/10.1016/j.jss.2017.08.031
25. Zhou, Z.Q., Xiang, S., and Chen, T.Y. Metamorphic testing for software quality assessment: A study of search engines. *IEEE Transactions on Software Engineering 42*, 3 (Mar. 2016), 264–284.

Zhi Quan Zhou (zhiquan@uow.edu.au) is an associate professor in software engineering at the School of Computing and Information Technology, University of Wollongong, Wollongong, NSW, Australia.

Liqun Sun (ls168@uowmail.edu.au) is pursuing an M.Phil. degree in computer science at the University of Wollongong, Wollongong, NSW, Australia, and a software engineer at Itree, Wollongong, Australia.

DOI:10.1145/3231588

The system transforms raw telemetric data into engaging and informative blog texts readily understood by all.

BY ADVAITH SIDDHARTHAN, KAPILA PONNAMPERUMA, CHRIS MELLISH, CHEN ZENG, DANIEL HEPTINSTALL, ANNIE ROBINSON, STUART BENN, AND RENÉ VAN DER WAL

Blogging Birds: Telling Informative Stories About the Lives of Birds from Telemetric Data

BLOGGING BIRDS IS a novel artificial intelligence program that generates creative texts to communicate telemetric data derived from satellite tags fitted to red kites—a medium-size bird of prey—as part of a species reintroduction program in the U.K. We address the challenge of communicating telemetric sensor data in real time by enriching it with meteorological and cartographic data, codifying ecological knowledge to allow creative interpretation of the behavior of individual birds in respect to such enriched data, and dynamically generating informative and engaging data-driven blogs aimed at the general public.

Geospatial data is ubiquitous in today's world, with vast quantities of telemetric data collected by GPS receivers on, for example, smartphones and automotive black boxes. Adoption of telemetry has been particularly striking in the ecological realm, where the widespread use of satellite tags has greatly advanced our understanding of the natural world.[14,23] Despite its increasing popularity, GPS telemetry involves the important shortcoming that both the handling and the interpretation of often large amounts of location data is time consuming and thus done mostly long after the data has been gathered.[10,24] This hampers fruitful use of the data in nature conservation where immediate data analysis and interpretation are needed to take action or communicate to a wider audience.[25,26]

The widespread availability of GPS data, along with associated difficulties interpreting and communicating it in real time, mirrors the scenario seen with other forms of numeric or structured data. It should be noted that the use of computational methods for data analysis per se is hardly new; much of science depends on statistical analysis and associated visualization tools. However, it is generally understood that such tools are mediated by human operators who take responsibility for identifying patterns in

» key insights

- **Summarizing environmentally enriched satellite-tag data in the form of informative, engaging, and fluent blogs is a challenge, even for trained ecologists, and computer-generated blogs were preferred by readers.**
- **Natural language generation, specifically data-to-text technology, is sufficiently advanced to achieve more than just factual summarization of data for professional use.**
- **This development opens new avenues for addressing societal challenges related to communicating data effectively and engaging the public with scientific research.**

IMAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

data, as well as communicating them accurately. An important but relatively recent addition to the growing field of data science is a technology called natural language generation[15] that automates the entire data pipeline to produce textual reports from data, whether numeric or structured. Originally developed to offer decision support in the workplace, natural language generation has generated textual summaries of technical data for professionals, including engineers, nurses, and oil-rig workers,[5,9,13,21] and is increasingly mainstream. Gartner, Inc. forecast in 2017 that 90% of business intelligence systems will incorporate natural language generation by 2019.[11] Companies like Arria (https://www.arria.com/), Narrative Science (https://narrativescience.com/), and Automated Insights (http://automatedinsights.com/) have developed software that summarizes data as textual reports; indeed, print-media organizations are increasingly turning to robo-journalism, and many routine data-driven news stories that are time consuming and mundane for professional journalists to write are being written entirely by computer programs. Such data-to-text applications require accuracy and clarity first and foremost, and it has been noted that for workplace applications consistency in language use is the main reason why computer-generated output is preferred to text produced by humans.[16]

At the other end of the spectrum of computer-generated language is the discipline of computational creativity, whereby computer programs attempt to construct jokes,[1] short stories,[7] and poetry.[8] Here, we use the term "creativity" in the context of "creative writing," defined by the *Oxford Dictionary* as "writing, typically fiction, or poetry, which displays imagination or invention (often contrasted with academic or journalistic writing)."[a] It is frequently said that creativity, especially in relation to design, requires the work to not just be imaginative or inventive but also "appropriate," as in Sternberg.[19] In his account of writing as design, Sharples[18] related the idea of appropriateness to "constraints," which provide the framework and context for creative expression and can be imposed either by the literary genre or by the conceptual space in which the writer is working.

Computer programs for computational creativity use static knowledge sources, typically manually construct-

[a] https://en.oxforddictionaries.com/definition/creative_writing

ed, to source joke templates, narrative plots, story grammars, and characters. In the storytelling domain, creativity manifests itself through emergent narratives dynamically created through the interactions of characters modeled as intelligent agents,[20] construction of different narratives from the same underlying plot representation,[17] or the tailoring of linguistic components to generate human-like narrative prose.[3] Deep neural networks have recently been applied to the generation of poetry by predicting likely word sequences fitting a mood or theme while also modeling tonal and structural constraints imposed by specific genres like Chinese quatrains.[28,29]

Earlier work exists on communicating spatiotemporal data in the form of stories[22] to help children with complex communication needs describe their school day to their parents. Here, microphone and radio frequency identification (RFID) readers were mounted on wheelchairs to make audio recordings by teachers or interactions with RFID-tagged locations, people, and objects. In this work, the computer-generated text was restricted to a factual summary of interactions recorded by RFID, while creativity was incorporated either through voice recordings provided by teachers or through functionality that allowed the children to personalize their stories by editing system output.

The body of work summarized earlier either generates factual reports from real-world data, with creativity introduced through direct human input[22] or generates creative texts from formal representation without recourse to real-world data. We are unaware of any previous computer program that generates creative texts from real-world data without human input. Addressing this gap, we describe Blogging Birds, which we designed to generate creative texts from data generated by satellite tags fitted to animals. The focal species for Blogging Birds is the red kite (*Milvus milvus*). This bird of prey was once widespread in the U.K., but prolonged and intense persecution led to its near extinction by the 1940s.

In 1989, the Royal Society for the Protection of Birds (RSPB) started a scheme to reintroduce the species in various locations across the U.K.[4] In one of these locations, the Black Isle near Inverness in the north of Scotland, several birds were equipped with solar-powered satellite tags. Limited human resources meant the tags were used mainly to locate birds that had died to foster detection and prosecution of possible wildlife crimes. However, it was felt there was scope for using data from these tags for public-engagement activities surrounding the reintroduction initiative, to communicate ecological insight that enhances people's understanding of the species, and to create a positive image of the species to harness public support for the reintroduction.[24] RSPB staff were themselves also keen to gain a better understanding of the lives of reintroduced birds, particularly how they recolonized a landscape that held precious few red kites for well over a century. They appreciated the inherent limitations in the data generated by the tags and were open to imaginative interpretations of the data, so long as the behaviors being narrated were ecologically plausible.

These requirements allowed us an opportunity to investigate data-driven generation of creative texts by computers, something we believe Blogging Birds is so far unique in its ability to achieve. The generated texts are creative in that they display imagination and inventiveness in how they interpret and report data under constraints imposed by kite ecology and the data itself. We sought to answer two research questions through experiments: Would the computer-generated blogs be well perceived by readers in comparison to blogs written by humans based on the same data?; and How important would the creative narration of ecological insight be to readers' perceptions of computer-generated blogs?

Figure 1. System architecture.

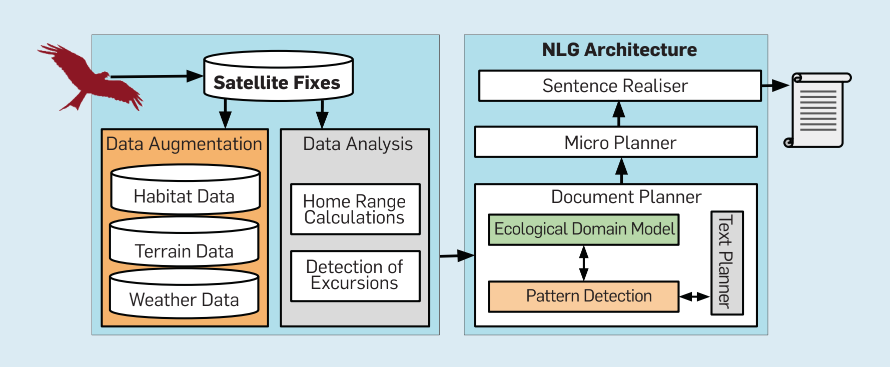


Table 1. Example augmented data used for pattern mining for one day of one week for a particular bird.

Day of Week	Hour	Habitat	Weather	Temp (C°)	Visibility (meters)	Wind Speed (miles per hour)	Location	Features	Distance Flown (miles)	Other Kites
Friday	08:00	Coniferous woodland	Overcast	13.0	24,000	3	East Croachy	Loch Ruthven	0	
Friday	10:00	Rough grassland	Heavy rain	13.9	5,000	2	Torness	Loch Ruthven	4	
Friday	12:00	Rough grassland	Heavy rain	16.0	3,600	2	Torness	Loch Ruthven	2	
Friday	14:00	Rough grassland	Heavy rain	16.0	3,600	2	Torness	Loch Ruthven	2	Merida
Friday	16:00	Improved grassland	Overcast	18.4	45,000	2	Torness		3	

The Blogging Birds System

The starting aim of Blogging Birds was to bring satellite-tagged individuals of a species (such as the red kite) "to life" by constructing ecologically sound narratives describing their movements. Conservationists fitted satellite tags—PTT-100 22-gram Solar Argos/GPS

PTT—to red kite chicks immediately prior to fledging, using a backpack harness designed for minimal hindrance. The tags were solar-powered and programmed to record up to six location fixes per day. Although this maximum could indeed be achieved during the summer months, a lack of sunlight in Scotland meant fewer fixes (a maximum of four per day) were obtained in spring and autumn and only the occasional fix during winter. To further preserve battery power, data was transmitted from the tag to the satellite only once per week. We thus configured Blogging Birds to produce a blog every week, or each time data was received from a bird.

Figure 1 outlines the overall architecture of the Blogging Birds system. We next describe the main components; see also Ponnamperuma et al.[12]

Data augmentation. The system processes an email messages with GPS fixes from the tags fixed to the red kites and enriches that data from readily available online sources about the local weather (https://www.metoffice.gov.uk/datapoint), habitat (such as different types of grassland and forests, https://eip.ceh.ac.uk/lcm), and geographic features (such as rivers, lochs, roads, and location names, https://www.ordnancesurvey.co.uk/). Table 1 presents a sample of the enriched data used by Blogging Birds.

Data analysis. The system then applies data-analysis procedures for identifying home ranges and patterns of movement with respect to these temporary settlement areas. Home ranges are identified as polygons using the Adehabitat package for R[2] by clustering the previous locations of an individual using 90% kernels. As described by van der Wal et al.,[24] we modeled local movement patterns as angular and radial velocity vectors to identify excursions, characterized by travel in relatively straight lines at higher speeds. This data analysis allows the document planner (described next) to detect the three prototypical patterns of movement in Figure 2, whereby the kite remains within a home range, explores an area outside its home ranges, or moves from one home range to another. Figure 3 shows the calculated home ranges for a bird (gray polygons), as well as the fixes classified as excursions (black crosses) and non-excursions (amber crosses).

Figure 2. Prototypical red kite movement patterns: C1 is small and constricted movements within an area of intense usage (home range); C2 is exploratory movement from a home range (round trip); and C3 is direct movements between separate home ranges.

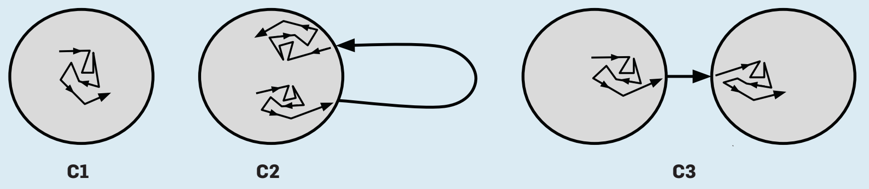
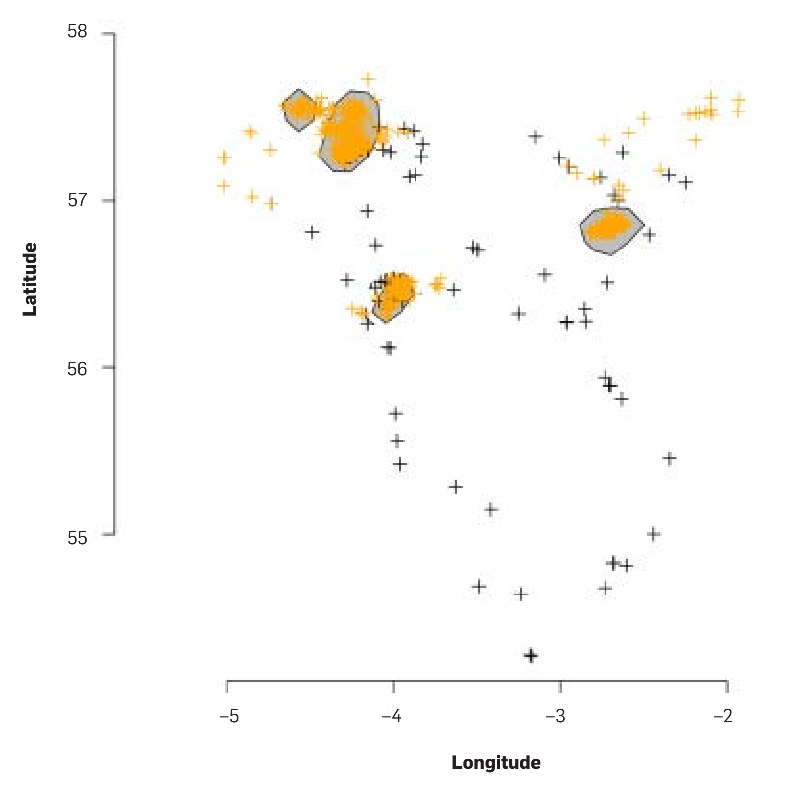


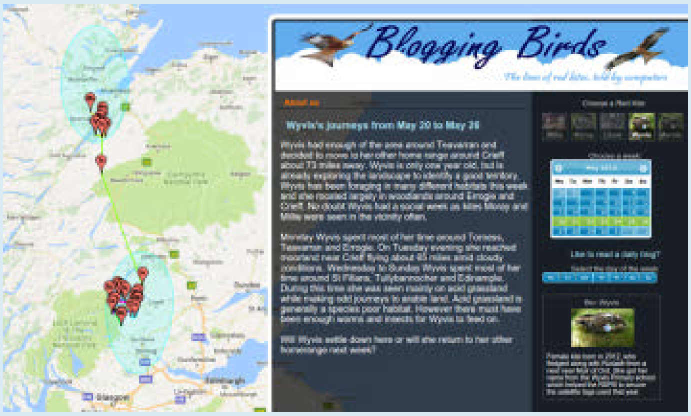
Figure 3. Calculated home ranges (gray polygons) and classification of fixes as excursions (black crosses) or non-excursions (amber crosses) for a particular bird.



Document planner. The document planner in Figure 3 identifies patterns in the data that signal different red kite behaviors and creates "messages" (implemented as Java classes) that encode these behaviors for use by the "micro planner" and "sentence realiser," which then generate sentences in English.

The data analysis allows us to detect the three prototypical patterns of movement outlined in Figure 2, whereby the kite remains within a home range, explores an area outside its home ranges, or moves from one home range to another. An ecological-domain model further defines different travel, foraging, and social behaviors as rules that can apply under specific environmental and geographic conditions; for instance, following heavy rain, a kite observed on any of the grassland habitats might feed on earthworms or a kite observed near a woodland habitat late in the afternoon is likely to be preparing to roost. These rules are implemented as JBoss Drools (http://www.jboss.org/drools), a business-logic-integration platform that allows us to instantiate messages when

Figure 4. Screenshot of the Blogging Birds Web interface.


Blogging Birds
The lives of red kites, told by computers
Choose a Red Kite
Choose a week
Like to read a daily blog?
Select the day of the week
Bio: Wyvis

About us

Wyvis's journeys from May 20 to May 26

Wyvis had enough of the area around Teavarran and decided to move to her other home range around Crieff about 73 miles away. Wyvis is only one year old, but is already exploring the landscape to identify a good territory. Wyvis has been foraging in many different habitats this week and she roosted largely in woodlands around Errogie and Crieff. No doubt Wyvis had a social week as kites Moray and Millie were seen in the vicinity often.

Monday Wyvis spent most of her time around Torness, Teavarran and Errogie. On Tuesday evening she reached moorland near Crieff flying about 65 miles amid cloudy conditions. Wednesday to Sunday Wyvis spent most of her time around St Fillans, Tullybannocher and Edinample. During this time she was seen mainly on acid grassland while making odd journeys to arable land. Acid grassland is generally a species poor habitat. However there must have been enough worms and insects for Wyvis to feed on.

Will Wyvis settle down here or will she return to her other homerange next week?

particular patterns are detected in the data. In total, the system implements Drools for 26 movement behaviors (such as flying along a coast or over a landmark like a castle or loch and the home-range-related movement patterns in Figure 2); 33 foraging behaviors, mostly detailing the food available for a kite in different habitats at different times of the year but also sometimes related to specific features (such as when a red kite near a road might be looking for roadkill); and six social behaviors (such as roosting and nesting); see the online appendix "Example Rules" (dl.acm.org/citation.cfm?doid=3231588&picked=formats).

The pattern-detection module then exhaustively applies the rules to the satellite fixes to produce a list of all observed movement behaviors and all possible foraging and social behaviors consistent with known environmental and geographic conditions. The latter is the first step in the creative process, whereby the program explores the conceptual space to "imagine" how the kite might have been behaving.

Blogging Birds uses a rule-based text planner for dynamic text generation. The planning rules decide how information is ordered, but what information to include and how to organize it into sentences is determined at runtime in a data-driven manner.

The blogs are always planned as three paragraphs, the first describing the overall trends, the second providing more detail on a day-to-day basis, and the third posing a question about what the kite might do next, as well as occasionally offering a conclusion.

The content is selected through a process of summarization and aggregation of information. This is the second creative aspect of the blog generation (the first involved imagining a wide range of possible behaviors), as it plans what story to tell from the imagined behaviors. Blogging Birds aims to provide an overview of the main behaviors and highlight aspects that might be interesting to the human reader. Movement behaviors are considered more interesting than foraging behaviors, and rarer foraging behaviors are prioritized over more frequent ones. Each blog attempts to inform the reader about different aspects of red kite ecology by selecting different behaviors from different days. The main steps are as follows:

Paragraph 1

Movement pattern. Generate a message based on the detected movement pattern—C1, C2, or C3 in Figure 2; if the age of the bird can be used to interpret this pattern, add such an interpretation message;

Habitats visited. Generate a message summarizing the habitats visited; and

Other kites. Generate a message about other kites recorded nearby, if any.

Paragraph 2

Days of the week. Iterate over each day of the week (Monday to Friday):

- If the bird remained relatively static—C1 in Figure 2—then generate a message about nearby places or generate a message about any movement behavior detected; and
- Generate a message about a new (not previously used) possible foraging behavior, if any deduced; unusual (historically infrequent) behaviors are selected over common ones.

Remove redundancy. Aggregate the messages generated for the week through these two steps to remove redundancy (such as by grouping together days with similar behaviors).

Paragraph 3

Movement pattern. Generate a message for a question or comment based on the movement pattern—C1, C2 or C3—with the aim of intriguing the reader.

Micro planner and sentence realiser. The micro planner takes the messages generated by the document planner, implements aggregation through a variety of linguistic devices (such as ranges, coordination, and subordination), and limits linguistic repetition by varying the vocabulary. It provides sentence specifications to the "sentence realiser," which then generates sentences using the SimpleNLG library.[6]

Figure 4 is a screenshot of the Blogging Birds interface in which an automatically generated weekly blog for a kite is overlaid on a Google map of the bird's whereabouts with its historical home ranges marked as blue polygons. In this example, Wyvis, one of five red kites being blogged about, has traveled between two home ranges (movement pattern C3), and an explanation for the observed movement pattern is provided based on the age

Table 2. Baseline computer-generated blog without reference to ecological concepts for the week outlined in Figure 4.

Wyvis had enough of the area around Teavarran and decided to move to Crieff approximately 73 miles away. No doubt Wyvis had a social week, as kites Moray and Millie were often seen in the vicinity.

Monday, Wyvis spent most of her time around Torness, Errogie, and Teavarran. On Tuesday evening, she reached moorland near Crieff flying approximately 65 miles amid cloudy conditions and averaging a remarkable 11 miles per hour. The next five days Wyvis spent most of her time around Edinample, Tullybannocher, and St Fillans. During this time, she was seen mainly on acid grassland while making occasional journeys to farmland.

Will Wyvis settle down here?

of the bird. The system emphasizes the social side with reference to roosting and encounters with other tagged kites. The second paragraph is narrated chronologically, with care taken again to emphasize any unusual behaviors (such as the long distance flown on Tuesday) and to reference weather conditions ("cloudy") to make the text more engaging. Information is also provided about the foraging potential of the different habitat types visited. Aggregation is used to avoid repetition, using linguistic devices (such as range "Wednesday to Sunday," coordination "St Filans, Tullybannocher, and Edinample," and subordination "mainly on acid grassland, while making odd journeys to arable land"). The question posed in the final paragraph is selected based on the movement pattern detected.

Here, we focus on situations where the timeframe covered by each blog is set at one week, as this is the frequency at which the tags are programmed to transmit data. However, the system architecture is sufficiently generic to be able to handle other timeframes, and the interface also allows the user to select a day of the week and read a blog composed for that day. Blogs could in theory also be provided for longer timeframes, but as the goal of the project was to allow readers to monitor or follow the birds on a continuous basis, this option was not implemented.

Evaluation

We investigated both how computer-generated blogs are appraised by readers in comparison to human-written blogs based on the same data and the contribution of the generated ecological insights to such appraisals. To this end, we designed studies to evaluate the quality of the computer-generated blogs for different patterns of movement, first through comparison with blogs written manually, then through comparison with baseline computer-generated blogs that report the data factually without ecological insights.

Method. We focused on the three prototypical movement patterns outlined in Figure 2 as conditions C1, C2, and C3. For each condition, we identified 12 weeks of data such that the focal red kite's movements broadly matched this condition (for example, the week in Figure 4 would correspond to C3), giving us 36 weeks of data in total.

The generated texts are creative in that they display imagination and inventiveness in how they interpret and report data under constraints imposed by kite ecology and the data itself.

Comparison with human-written blogs. We recruited 12 post-graduate master's-level ecology students from the University of Aberdeen in Scotland (representative of those who might be hired by a conservation charity) to take part in a two-hour session on "digital media in nature conservation" outside teaching hours. We told them they would be writing three short blogs on the basis of environmental data we would provide, saying it would take them approximately 1.5 hours, that partaking would benefit our research while giving them unique insights into new technologies, and we would compensate them £15 cash to express our gratitude for helping us while learning.

We provided each writer with access to a one-page information sheet about red kites that summarized the typical movement patterns and foraging and social behaviors that were encoded in the Blogging Birds system. They were also free to consult any online sources they preferred. We also provided them with the enriched data available to the system for the week, presented in both tabular form (as in Table 1) and overlaid on a map showing home ranges and fixes (as in Figure 4, but without the blog). The information we provided to the 12 student writers was sufficient to allow them to make the same inferences as the system. However, in order to grant full creative freedom to the writers and avoid priming them to write similar blogs to the system, we avoided giving them direct access to the inferences made or used by the system. They were further informed about the intended purpose of the blogs and the target audience, and each was asked to write three 200-word blogs; that is, for data from three different weeks, one in each condition (C1–C3 in Figure 2) such that for each of the 36 weeks selected for the study we had one manually written blog. The order in which writers encountered each condition was randomized and writers not made explicitly aware of the existence of these conditions in the study, though the patterns were clearly visible on the respective maps and described on the information sheet. These 36 manually written blogs were compared to computer-generated blogs for the same weeks in the evaluation.

As our goal was to investigate Blogging Birds not just as a tool for those with an interest in nature conservation but as a resource to engage those interested in new technologies. We ran evaluations with two distinct groups of participants: 93 undergraduate biology students enrolled in a second year "community ecology" course and 49 first- and second-year undergraduates from across disciplines enrolled in a course entitled "digital society," both at the University of Aberdeen. In each trial, a participant sitting at an individual workstation was shown an interface with a map with home ranges and fixes of a kite for one of the weeks, as well as two blogs, one written manually and one computer-generated, without any information about their provenance. Participants said what blog they preferred (or expressed no preference) and also rated each blog on how informative, fluent, and engaging they found it on a seven-point Likert scale. Each participant evaluated three pairs of blogs. We designed the study to test three specific hypotheses:

H1. Computer-generated blogs are preferred to human-written blogs;

H2. Computer-generated blogs are more informative, fluent, and engaging than human-written blogs; and

H3. The differences in ratings for computer-generated and human-written blogs are conditional on the movement pattern of an individual bird C1, C2, or C3, as in Figure 2.

Comparison with baseline. To directly evaluate whether communicating ecological insights through the blogs is important to readers, we compared Blogging Birds to a computer-generated baseline that blogs about the movement patterns without reference to ecological concepts; see Table 2 for an example. These baseline blogs were entirely factual and reported behaviors only directly observed in the data but that otherwise followed the same format as the full-system blogs. An additional 27 undergraduate students enrolled in the digital society course, but who had not participated in the earlier experiment, evaluated the full vs. the baseline system using the same methodology and interface as before. We designed this study to test two specific hypotheses:

H4. Computer-generated blogs with ecological insights are preferred to baseline computer-generated blogs without ecological insights; and

H5. Computer-generated blogs with ecological insights are more informative and engaging than baseline computer-generated blogs without ecological insights, while their fluency is comparable.

Figure 5. Preferences for human-written and for computer-generated blogs by movement condition, as in Figure 2: C1 is movement within a home range; C2 is a round trip; and C3 is movement between home ranges.

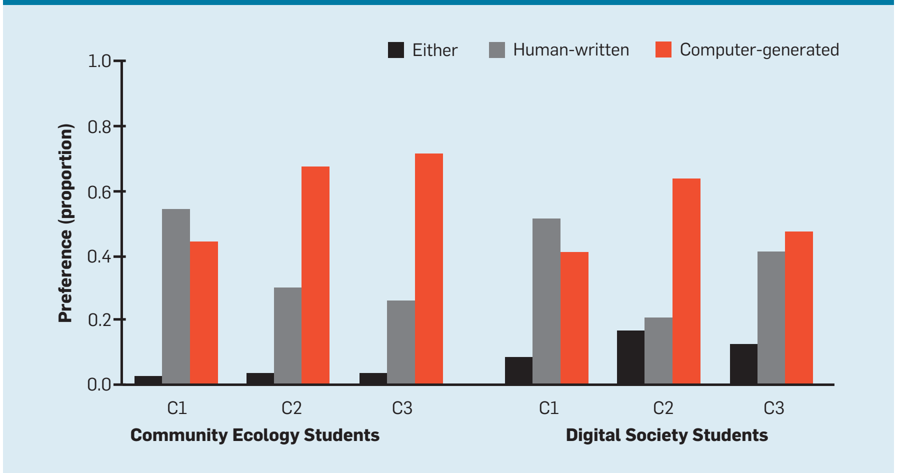
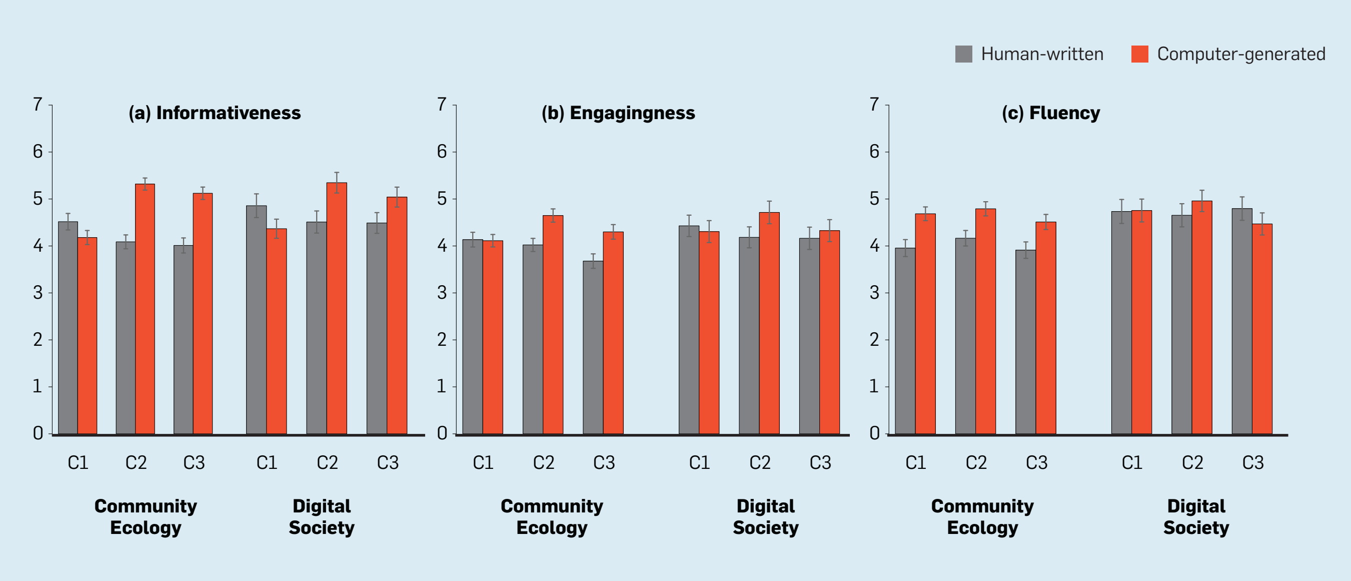


Figure 6. Average ratings for human-written and for computer-generated blogs by movement condition, as in Figure 2: C1 is movement within a home range; C2 is a round trip; and C3 is movement between home ranges.



Results

Evaluation against human-written blogs. Both sets of students showed an overall significant preference for the computer-generated blogs (238 trials vs. 153 trials in which human-written blogs were preferred; $\chi^2 = 18.5$; $p < 0.001$), confirming hypothesis H1. However, a more complex pattern emerged (see Figure 5), with this preference being dependent on the type of kite movement covered in the blog—C1, C2, or C3—and the orientation of the course—ecology or technology.

Across the community ecology students, there was a strong preference for computer-generated blogs when they captured more extensive movement by the kites—round trips (C2) and movement between home ranges (C3)—while there was little difference in preference between the two blog types when kite movement was limited; that is, small movements within home ranges (C1). Digital society students showed an overall clear preference for the computer-generated blogs only when they described round trips (C2). Combined, our findings indicate Blogging Birds is particularly skilled at handling cases where the focal bird shows substantial movement. Average ratings for how fluent, engaging, and informative the blogs were (see Figure 6) showed the main perceived advantage of the computer-generated blogs pertains to their "informativeness," with smaller improvements visible for how engaging and fluent they were.

We ran a MANOVA, with informativeness, engagingness, and fluency as the dependent variables and blog type (computer or human), kite movement-pattern (C1, C2, or C3), student group (community ecology or digital society), and their interactions as fixed effects, and writer ID and evaluator ID as random effects. We found the following main effects and interactions at $p<0.01$: computer-written blogs were rated significantly higher ($p<0.0001$) than human-written blogs (confirming hypothesis H2); students in the digital society course gave higher ratings overall than students in community ecology ($p<0.01$); and there was interaction between blog type and movement pattern ($p<0.0001$), confirming hypothesis H3. Post-hoc analysis using the Tukey HSD test on the individual ANOVAs with Bonferroni-correction revealed this interaction came about because the computer-generated blogs capturing conditions with more movement by kites (C2 and C3) were more informative than the human-written blogs for the same conditions and more informative than computer-generated blogs capturing constricted movement (C1) ($p<0.0001$ for each comparison).

To better understand these described effects, we compared the distribution of ratings obtained by each human writer (H1–H12) and the computer (Comp) in Figure 7. Only two of the blog writers (H3 and H10) were deemed to write more informative blogs than the computer, and both of them were considered less engaging and fluent than the computer-generated blogs. Likewise, H4, who wrote more fluent and engaging blogs than the computer, was rated rather low

Figure 7. Computer-generated blogs (Comp) vs. human-written blogs (H1–H12).

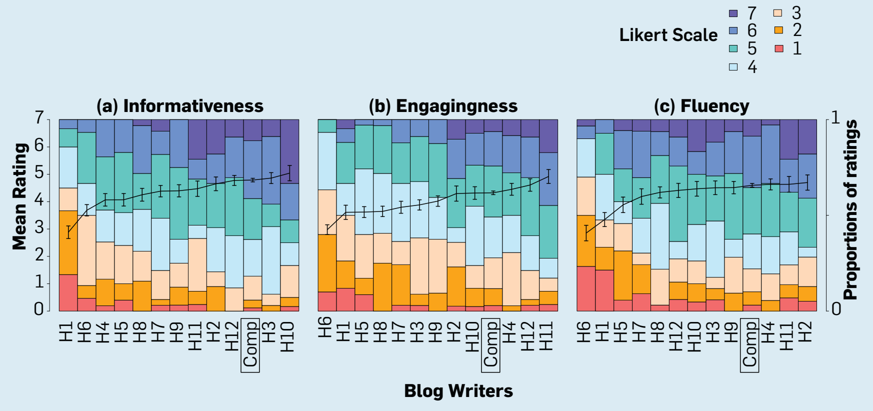
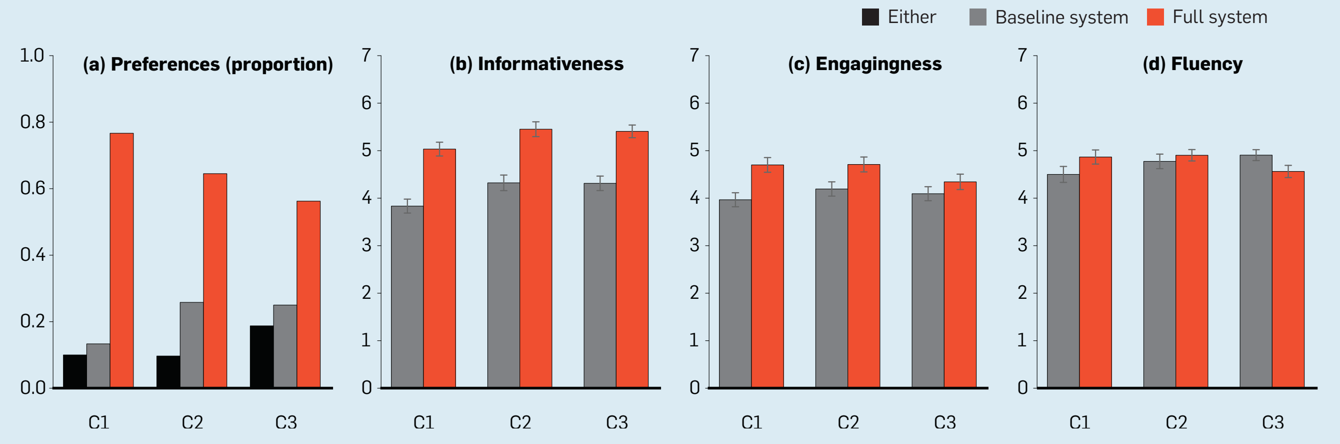


Figure 8. Computer-generated blogs with ecological insights (full system) vs. computer-generated blogs describing movement patterns only (baseline system).



for "informativeness," thus illustrating the difficulty of being informative, engaging, and fluent at the same time, even for a human writer. Indeed, all the writers were committed and used the full 1.5 hours for composing the blogs, yet most were outperformed by the computer on each of the three metrics. For examples of human-written and computer-generated blogs, as well as details of how they were appraised by evaluators, see the online appendix.

A questionnaire filled out by the blog writers provided many interesting insights. In general, they found it difficult to comprehend and summarize the sheer amount of data in fewer than 200 words but also felt the process became easier the more they did. There was, however, concern from many that the blogs were becoming repetitive, especially if there was little variation in what the red kites were actually doing, stemming largely from a lack of knowledge of kite ecology and behavior. Summarizing the range of data in different formats was certainly challenging, and some enjoyed the process more than others. There was considerable variability in how the blog writers used the materials provided them to create the blogs. Some concentrated mostly on the visible patterns on Google maps, others looked in more detail at the map data by clicking on individual map points to find out more, and yet others found inspecting the data in a tabular format was most useful. Asked whether they would like to write the red kite blogs as a job, the consensus was that, although initially enjoyable, it would quickly get tedious and increasingly more difficult to write non-repetitive material.

Evaluation against baseline. Participants demonstrated a conclusive preference for the full system with ecological insights, preferring it in 61 trials compared to only 20 trials in which the baseline was preferred ($\chi^2 = 21.5$; $p < 0.001$), confirming hypothesis H4. Interestingly, this effect was strongest when blogs described situations with little movement by the birds during those weeks (C1); here, the full-system blogs were preferred in 23 trials compared to just four baseline blogs ($\chi^2 = 13.4$; p=0.0002). For C2 and C3, the corresponding values were preferences for the full system in 20 and 18 trials, compared to preferences for the baseline in eight trials each (χ^2 = 5.1, 3.8; p = 0.0233, 0.0499). The absence of ecological interpretation by the baseline system was thus judged adversely for all movement patterns, particularly so when the birds were relatively static. We also found the full blogs were rated as more informative (p<0.0001) and more engaging (p=0.0215) but not more fluent (p=0.825) (see Figure 8), confirming hypothesis H5.

Telemetric data is ubiquitous, captured by smartphones and other mobile devices, as well as through GPS sensors embedded in vehicles used by the transportation industry and others.

The two studies we have presented here demonstrate that computer-generated blogs are appraised more positively than human-written blogs, and that computer-generated blogs with creatively generated ecological insights are preferred overwhelmingly to blogs generated from the same data but without inclusion of these insights.

Conclusion

The Blogging Birds system shows that raw satellite tag data can be transformed into fluent, engaging, and informative texts directed at members of the public and in support of nature conservation.

We demonstrated that computers can compete with human experts in generating creative stories from numerical data. Unlike natural language generation systems that generate texts for news reporting or for decision making in the workplace, Blogging Birds's narratives are not entirely factual. Though the system is constrained by the observed data and its ecological domain model, the red kites' reported foraging and social behaviors are only *imagined* to have taken place. However, including these behaviors in the narratives allows us to communicate red kite ecology to the reader, and the blogs are better appraised as a consequence. Our work thus simultaneously addresses the societal challenges of communicating data effectively and engaging the general public with scientific research.

Blogging Birds composes blogs by combining texts produced through three different types of analysis: The first is a generic factual summarization of telemetric data enriched with location-specific information about weather conditions, habitat type, and geographic features, and can be readily adapted for use in other domains. The second is the processing and ecological interpretation of movement data in the context of home range use, and the third is the exploitation of domain knowledge encoded as a collection of rules that help the system imagine possible foraging and social behaviors from environmental and geographic parameters. Much of what is creative and interesting about the blogs derives from the latter domain-specific types of data analyses. Although the developed principles apply more broadly, new applications would require construction of knowledgebases pertinent to the domain of use. While this is a clear limitation of our approach, note our ecological interpretation of movement data in particular would be applicable to several other species. For example, we have already developed a version of Blogging Birds for golden eagles (*Aquila chrysaetos*) for use by RSPB conservation officers, successfully reusing the second, as well as the first, type of analysis.

During the course of the project, we also discovered ecologists had limited knowledge of the foraging behavior of red kites in Scotland, as they had not been studied extensively following their relatively recent reintroduction. We could thus encode only a limited number of rules per habitat type. The absence of any large-scale corpus of texts in this domain also meant we could not apply the deep learning methods that are rapidly gaining popularity for generating linguistic variation in computer-generated texts.[27] In future work, we plan to invite Blogging Birds' users to contribute behavioral observations from across the U.K., enabling us to simultaneously curate a larger set of rules and further public engagement.

Finally, our ideas demonstrated here are applicable more generally. Telemetric data is ubiquitous, captured by smartphones and other mobile devices, as well as through GPS sensors embedded in vehicles used by the transportation industry and others. Even albums of time-stamped and geotagged photos provide data similar to what we used here. The nature of the blogs, along with the information sources used for data enrichment, would depend on the application, to blog about a holiday or reveal the provenance and journey of a food item in a supermarket. In effect, we have demonstrated it is possible to blog about such data through a process of data enrichment and natural language generation, opening up new avenues for using AI to engage people through data.

Acknowledgments

This research was supported by an award from the RCUK Digital Economy Programme to the dot.rural Digital Economy Hub, award reference EP/G066051/1.

References

1. Binsted, K. and Ritchie, G. Computational rules for generating punning riddles. *International Journal of Humor Research 10*, 1 (July 1997), 25–76.
2. Calenge, C. The package 'adehabitat' for the R software: A tool for the analysis of space and habitat use by animals. *Ecological modelling 197*, 3 (Apr. 2006), 516–519.
3. Callaway, C.B. and Lester, J.C. Narrative prose generation. *Artificial Intelligence 139*, 2 (Aug. 2002), 213–252.
4. Carter, I. *The Red Kite*. Arlequin Press, Chelmsford, Essex, U.K., 2007.
5. Gatt, A., Portet, F., Reiter, E., Hunter, J., Mahamood, S., Moncur, W., and Sripada, S. From data to text in the neonatal intensive care unit: Using NLG technology for decision support and information management. *AI Communications 22*, 3 (third quarter 2009), 153–186.
6. Gatt, A. and Reiter, E. SimpleNLG: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation* (Athens, Greece, Mar. 30–31). Association for Computational Linguistics, Stroudsburg, PA, 2009, 90–93.
7. Gervás, P. Computational approaches to storytelling and creativity. *AI Magazine 30*, 3 (Fall 2009), 49–62.
8. Ghazvininejad, M., Shi, X., Choi, Y., and Knight, K. Generating topical poetry. In *Proceedings of Empirical Methods in Natural Language Processing* (Austin, TX, Nov. 1–5). Association for Computational Linguistics, Stroudsburg, PA, 2016, 1183–1191.
9. Goldberg, E., Driedger, N., and Kittredge, R.I. Using natural language processing to produce weather forecasts. *IEEE Expert 9*, 2 (Apr. 1994), 45–53.
10. Hebblewhite, M. and Haydon, D.T. Distinguishing technology from biology: A critical review of the use of GPS telemetry data in ecology. *Philosophical Transactions of the Royal Society of London B: Biological Sciences 365*, 1550 (July 2010), 2303–2312.
11. Panetta, K. *Neural Networks and Modern BI Platforms Will Evolve Data and Analytics*. Gartner, Inc., Stamford, CT, Jan. 16, 2017; http://www.gartner.com/smarterwithgartner/nueral-networks-and-modern-bi-platforms-will-evolve-data-and-analytics/
12. Ponnamperuma, K., Siddharthan, A., Zeng, C., Mellish, C., and Wal, R. Tag2Blog: Narrative generation from satellite tag data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (Sofia, Bulgaria, Aug. 4–9). Association for Computational Linguistics, Stroudsburg, PA, 2013, 169–174.
13. Portet, F., Reiter, E., Gatt, A., Hunter, J., Sripada, S., Freer, Y., and Sykes, C. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence 173*, 7–8 (May 2009), 789–816.
14. Pschera, A. *Animal Internet: Nature and the Digital Revolution*. New Vessel Press, New York, 2016.
15. Reiter, E. and Dale, R. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, U.K., 2000.
16. Reiter, E., Sripada, S., Hunter, J., Yu, J., and Davy, I. Choosing words in computer-generated weather forecasts. *Artificial Intelligence 167*, 1–2 (Sept. 2005), 137–169.
17. Rishes, E., Lukin, S.M., Elson, D.K., and Walker, M.A. Generating different story tellings from semantic representations of narrative. In *Proceedings of the International Conference on Interactive Digital Storytelling* (Istanbul, Turkey, Nov. 6–9) Springer, New York, 2013, 192–204.
18. Sharples, M. An account of writing as creative design. In *The Science of Writing*. Lawrence Erlbaum, Hillsdale, NJ, 1996.
19. Sternberg, R.J. *Handbook of Creativity*. Cambridge University Press, Cambridge, U.K., 1999.
20. Theune, M., Faas, S., Heylen, D.K.J., and Nijholt, A. The virtual storyteller: Story creation by intelligent agents. In *Proceedings of the Conference on Technologies for Interactive Digital Storytelling and Entertainment*, S. Göbel et al., Eds. (Darmstadt, Germany, Mar. 24–26). Fraunhofer IRB Verlag, Stuttgart, Germany, 2003, 204–215.
21. Theune, M., Klabbers, E., de Pijper, J.-R., Krahmer, E., and Odijk, J. From data to speech: A general approach. *Natural Language Engineering 7*, 1 (Mar. 2001), 47–86.
22. Tintarev, N., Reiter, E., Black, R., Waller, A., and Reddington, J. Personal storytelling: Using natural language generation for children with complex communication needs, in the wild. *International Journal of Human-Computer Studies 92* (Aug. 2016), 1–16.
23. Tomkiewicz, S.M., Fuller, M.R., Kie, J.G., and Bates, K.K. Global positioning system and associated technologies in animal behaviour and ecological research. *Philosophical Transactions of the Royal Society of London B: Biological Sciences 365*, 1550 (July 2010), 2163–2176.
24. van derWal, R., Zeng, C., Heptinstall, D., Ponnamperuma, K., Mellish, C., Ben, S., and Siddharthan, A. Automated data analysis to rapidly derive and communicate ecological insights from satellite-tag data: A case study of reintroduced red kites. *Ambio 44*, 4 (Oct. 2015), 612–623.
25. Verma, A., van der Wal, R., and Fischer, A. Microscope and spectacle: On the complexities of using new visual technologies to communicate about wildlife conservation. *Ambio 44*, 4 (Oct. 2015), 648–660.
26. Wall, J., Wittemyer, G., Klinkenberg, B., and Douglas-Hamilton, I. Novel opportunities for wildlife conservation and research with real-time monitoring. *Ecological Applications 24*, 4 (June 2014), 593–601.
27. Wen,, T.-H., Gašić, M., Mrkšić, N., Su, P.-H., Vandyke, D., and Young, S. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Lisbon, Portugal, Sept. 17–21). Association for Computational Linguistics, Stroudsburg, PA, 2015.
28. Yan, R. I, Poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. In *Proceedings of the International Joint Conference on Artificial Intelligence*. New York, July 9–15). AAAI Press, Palo Alto, CA, 2016, 2238–2244.
29. Zhang, X. and Lapata, M. Chinese poetry generation with recurrent neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Doha, Qatar. Oct. 25–29). Association for Computational Linguistics, Stroudsburg, PA, 2014, 670–680.

Advaith Siddharthan (advaith.siddharthan@open.ac.uk) is a Reader in the Knowledge Media Institute at The Open University, Milton Keynes, U.K.

Kapila Ponnamperuma (kapila.ponnamperuma@arria.com) is the lead natural language engineer at Arria NLG plc, Aberdeen, Scotland, U.K.

Chris Mellish (c.mellish@abdn.ac.uk), now retired, was a professor of computer science at the University of Aberdeen, Scotland, U.K., at the time this research was conducted.

Cheng Zeng (zengc@hotmail.co.uk) was a research assistant on the Blogging Birds Project at the time this research was conducted.

Daniel Heptinstall (djheptinstall@gmail.com) is a senior international biodiversity adviser on the U.K. government's Joint Nature Conservation Committee.

Annie Robinson (annierobinson@abdn.ac.uk) was a Research Fellow on the Blogging Birds Project at the time this research was conducted.

René van der Wal (r.vanderwal@abdn.ac.uk) is a professor of ecology at the University of Aberdeen, Scotland, U.K.

DOI:10.1145/3226588

A new model for describing the Internet reflects today's reality and the future's needs.

BY PAMELA ZAVE AND JENNIFER REXFORD

The Compositional Architecture of the Internet

IN 1992, THE explosive growth of the World Wide Web began. The architecture of the Internet was commonly described as having four layers above the physical media, each providing a distinct function: a "link" layer providing local packet delivery over heterogeneous physical networks, a "network" layer providing best-effort global packet delivery across autonomous networks all using the Internet Protocol (IP), a "transport" layer providing communication services such as reliable byte streams (TCP) and datagram service (UDP), and an "application" layer. In 1993, the last major change was made to this classic Internet architecture;[11] since then the scale and economics of the Internet have precluded further changes to IP.[12]

A lot has happened in the world since 1993. The overwhelming success of the Internet has created many new uses and challenges that were not anticipated by its original architecture:

- ▸ Today, most networked devices are mobile.
- ▸ There has been an explosion of security threats.
- ▸ Most of the world's telecommunication infrastructure and entertainment distribution has moved to the Internet.
- ▸ Cloud computing was invented to help enterprises manage the massive computing resources they now need.
- ▸ The IPv4 32-bit address space has been exhausted, but IPv6 has not yet taken over the bulk of Internet traffic.
- ▸ In a deregulated, competitive world, network providers control costs by allocating resources dynamically, rather than provisioning networks with static resources for peak loads.

Here is a conundrum. The Internet is meeting these new challenges fairly well, yet neither the IP protocol suite nor the way experts describe the Internet have changed significantly since 1993. Figure 1 shows the headers of a typical packet in the AT&T backbone,[19] giving us clear evidence that the challenges have been met by mechanisms well outside the limits of the classic Internet architecture. In the classic description, the only headers between

» key insights

- For the past 25 years, the Internet has been evolving to meet new challenges by composition with new networks that were unanticipated by the original architecture. New networks make alternative trade-offs that are not compatible with the general-purpose Internet design, or maintain alternative views of network structure.
- In an architecture composed of self-contained networks, each network is a microcosm with all the basic network mechanisms including a namespace, routing, a forwarding protocol, session protocols, and directories. The mechanisms are specialized for the network's purposes, membership, geographical span, and level of abstraction.

IMAGE BY TIERNEYMJ

HTTP and Ethernet would be one TCP header and one IP header.

In this article, we present a new way of describing the Internet, better attuned to the realities of networking today, and to meeting the challenges of the future. Its central idea is that the architecture of the Internet is a flexible composition of many networks—not just the networks acknowledged in the classic Internet architecture, but many other networks both above and below the public Internet in a hierarchy of abstraction. For example, the headers in Figure 1 indicate the packet is being transmitted through six networks below the application system. Our model emphasizes the interfaces between composed networks, while offering an abstract view of network internals, so we are not reduced to grappling with masses of unstructured detail. In addition, we will show that understanding network composition is particularly important for three reasons:

Reuse of solution patterns: In the new model, each composable network is a microcosm of networking, with the potential to have all the basic mechanisms of networking such as a namespace, routing, a forwarding protocol, session protocols, and directories. Our experience with the model shows this perspective illuminates solution patterns for problems that occur in many different contexts, so that the patterns (and their implementations!) can be reused. This is a key insight of Day's seminal book *Patterns in Network Architecture*.[7] By showing that interesting networking mechanisms can be found at higher levels of abstraction, the new model helps to bridge the artificial and unproductive divide between networking and distributed systems.[17]

Verification of trustworthy services: Practically every issue of *Communications* contains a warning about the risks of rapidly increasing automation, because software systems are too complex for people to understand or control, and too complex to make reliable. Networks are a central part of the growth of automation, and there will be increasing pressure to define requirements on communication services and to verify they are satisfied.[14] As we will show, the properties of trustworthy services are defined at the interfaces between networks, and are usually dependent on the interaction of multiple networks. This means they cannot be verified without a formal framework for network composition.

Evolution toward a better Internet: In response to the weaknesses of the current Internet, many researchers have investigated "future Internet architectures" based on new technology and "clean slate" approaches.[2,20,21,25] These architectures are not compatible enough to merge into one network design. Even if they were, it is debatable whether they could satisfy the demands for specialized services and localized cost/performance trade-offs that have already created so much complexity. A study of compositional principles and compositional reasoning might be the key to finding the simplest Internet architecture that can satisfy extremely diverse requirements.

We begin with principles of the classic architecture, and then discuss why they have become less useful and how they can be replaced. This should help clarify that we are proposing a really new and different way of talking about networks, despite the familiarity of the terms and examples. We close by considering potential benefits of the new model.

The User Interface to a Network

The end-to-end principle. The best-known principle of the classic Internet architecture is the end-to-end principle,[5,8] which creates a sharp divide between the network and the endpoint machines that it serves. The principle says the functions of the network should be minimized, so that it serves everyone efficiently, and that whenever possible services should be implemented in the endpoint machines. The endpoints are easily programmable (so anyone can add services), and the end-to-end perspective is the best perspective for functions such as reliability.

The end-to-end principle is also expressed by the slogan "smart edge, dumb network." Another implication of the end-to-end principle is the user interface to a network consists of the links between endpoint machines and the rest of the network.

Despite its tremendous explanatory and engineering value, the end-to-end principle does not describe the Internet as a whole. We know there are services (such as protection from denial-of-service attacks) that cannot be implemented in endpoints.[4] Today's Internet is full of *middleboxes*, which are functional elements located inside the network and inserted into end-to-end paths. On the other side of the divide, performance of the network depends to some extent on congestion control in TCP endpoints. It is no longer true that endpoint machines and networks are always owned by different parties (in clouds they are not), and no longer true that network elements such as routers are not programmable.[10]

From a modeling perspective, the divide between network and endpoints is harmful for a very simple reason: If we want to describe and verify communication (network) services, then we must include all the agents involved in providing those services.

User interfaces are inside machines. Figure 2 illustrates the new model's approach to network services and the user interface. Each machine participating in a network must be running a *member* of that network. The network member is a software or hardware module that implements some subset of the network protocols. Members are connected by links, where a link is a communication channel that accepts packets from one member and delivers them to another member.[a] Members of the network forward packets that are not destined for them, so a packet can reach its destination through a *path* of members.

The users of networks are distributed application systems—computer systems with operational modules spread across different physical machines. The modules of a distributed system need a network to communicate. The main user interface to a network consists of the interfaces inside machines between user modules and members of the network.

An instance or usage of network service is a *session*. A network has *packets*, which are its transmissible units of data. A session transmits a set of packets that are related from the perspective of the user. In Figure 2, a one-way session transmits packets from an ap-

[a] Although the model allows one-to-many sessions and links, for services such as broadcast and multicast, they are omitted for simplicity.

plication sender to an application receiver. The session has identifier *sessIdent*. So the main user interface to the network is that the sender has action *send* (*packet*, *sessIdent*) to send a packet in the session, and the network has action *deliver* (*packet*, *sessIdent*) to deliver a packet to the receiver.

Although the user interface between two networks is always implemented inside machines, implementations vary. Many user interfaces are implemented by software in operating systems. The user interfaces to the MPLS networks in Figure 1, on the other hand, are implemented deep inside the hardware of high-speed routers.

The Nature of a Layer

Fixed layers with distinct functions. The classic Internet architecture prescribes five layers (including the physical media), as listed earlier. The contemporaneous OSI reference model[13] has seven layers, with "session" and "presentation" layers between the transport and application layers. In both hierarchies each layer has a distinct function not performed by any other layer.

Fixed layers with distinct functions are no longer a realistic description of the Internet. For example, routing and forwarding are extremely important network functions; in the classic architecture the local version of these functions resides in the link layer, while the global version resides in the network layer. Yet, Figure 1 also shows the presence of a GPRS (a standard for cellular data service) network and two MPLS networks, each of which has its own routing and forwarding that aggregates packets and manages resources at different levels of abstraction. Further up in Figure 1, we see three IP headers, plus evidence that three separate IP session protocols (TCP, IPSec, and UDP) apply to this packet.

Conceivably there is a model with fixed layers and distinct functions that fits this packet, but the same HTTP message—if observed at different places along its end-to-end path—will be encapsulated in packets with different headers indicating different layers and different functions. So no variation on the classic Internet architecture or OSI reference model can help us understand what is going on.

Self-contained networks. A major principle of the new model is that layers in a composition hierarchy are self-contained networks. Each network is a microcosm of networking, with all the basic mechanisms including a namespace, routing, a forwarding protocol, session protocols, and directories. However, because networks vary widely in their purposes, geographical spans, memberships, and levels of abstraction, these mechanisms also vary, and a mechanism may be vestigial in a particular network design where it is not needed. According to this principle the IP protocol suite is a general-purpose network design that is implemented on most networked devices. As such, it can be reused for the design and implementation of many networks. Note that an IP network encompasses both the network and transport layers of the classic Internet architecture.

We will now give brief explanations of the major parts of a network, followed by examples.

A network's *namespace* is the set of names that its members can have. Most commonly each member of a network has a unique name, although there are many exceptions.

Routing is the mechanism that determines paths and installs entries in the forwarding tables of network members, while a network's *forwarding protocol* is the mechanism in which a member uses its forwarding table and other computations to forward packets toward their destinations. It includes formats for packet headers and forwarding tables. Most common-

Figure 1. Headers of a typical packet in the AT&T backbone network.

Headers lower in the diagram are outermost in the actual packet.

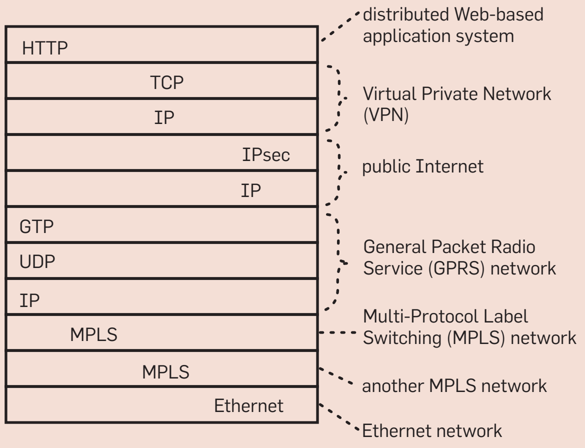
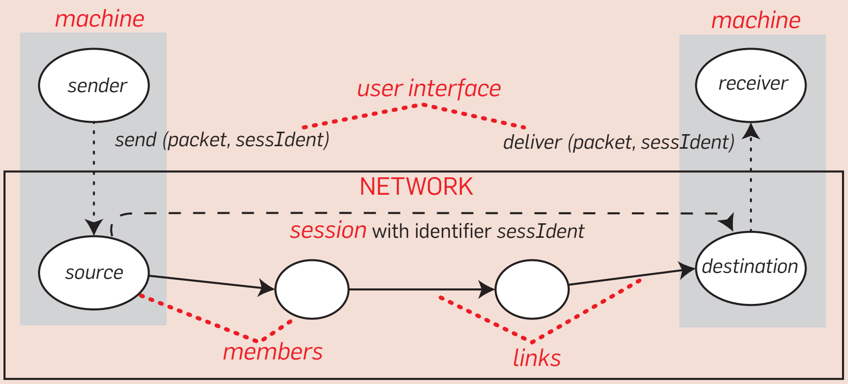

HTTP
TCP
IP
IPsec
IP
GTP
UDP
IP
MPLS
MPLS
Ethernet
distributed Web-based application system
Virtual Private Network (VPN)
public Internet
General Packet Radio Service (GPRS) network
Multi-Protocol Label Switching (MPLS) network
another MPLS network
Ethernet network

Figure 2. The main user interface to a network, with an example session.



ly, the table at each member is a mapping from *headerPattern* and *inLink* to *outLink*, where *headerPattern* matches some subset of packet headers, and *inLink* and *outLink* are local identifiers for the links of that member. The mapping tells the member that on receiving a packet on incoming link *inLink* whose header matches *headerPattern*, it should forward the packet onto outgoing link *outLink*. The mapping can also tell the member, explicitly or implicitly, to drop the packet.

A *session protocol* is a set of conventions governing a specific kind of session; it always includes the behavior of the session endpoint members, and may include the behavior of other network members on the session path. It covers packet headers, packet sequence, member state, and member actions. The header format of a session protocol is a specialization of its network's forwarding format, so a header must conform to both. The new model makes particular use of the following header fields:

- the name of the *destination* endpoint;
- a *session protocol identifier*;
- a *session identifier*;
- a *user network* to identify the network being served by the session (as we will discuss).

These fields are always present in headers unless they are vestigial (which means they would be identifying elements in a set of size zero or one) or unless the information they carry is already stored in members along the path of the session.

Examples of new networks. Many campus architectures have networks called virtual local area networks (VLANs) that are not found in the classic architecture.[22] The purpose of VLANs is to maintain an important network topology that is not present in either the IP network or local area networks (LANs) on campus, as shown in Figure 3. In the figure, each physical machine is assigned a color and final name digit for its network members, so that it is easy to see which network members are on the same machine.

At the bottom level we see there are physical LANs covering different areas of campus, and some high-speed physical links across campus.[b] At the top level the campus has a private IP network. User machines are divided into groups depending on whether they belong to students, administrators, departments, or others. Members of a group are identifiable by the prefixes of their IP addresses (abbreviated in the figure). Within each group each user machine is connected by a virtual link to every other group member and to one or more IP routers that serve as security gateways to the group. Members of different groups can reach each other only through IP routers, where filtering rules are installed to allow only approved communication among groups. Note that the machines with IP addresses 2.7 and 2.8 are close together in the IP topology, but far apart in the physical topology.

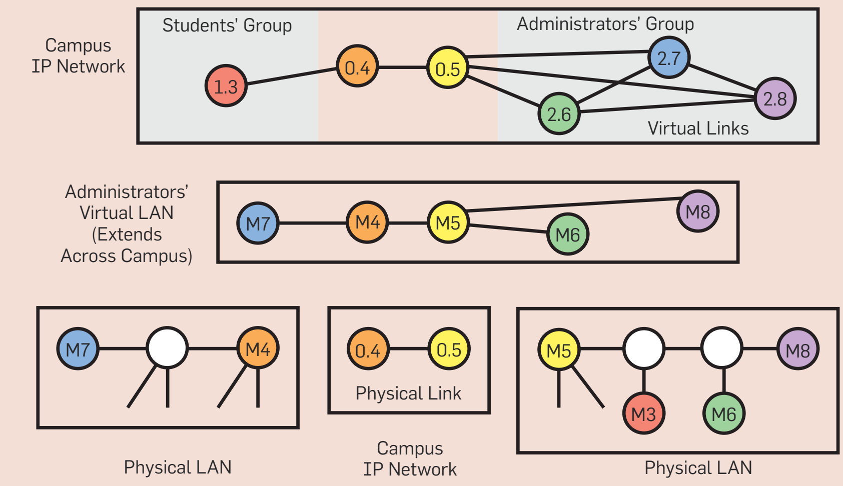
At the middle level of the figure there is an isolated VLAN for each group. Like the LANs, a VLAN uses the Ethernet design in which names are MAC addresses (abbreviated "M"). These VLANs do their own routing, separate from the routing in the LANs. A virtual link in the IP network must be implemented by a path in a VLAN and a link in a VLAN must be implemented by a physical path. As a result, a packet from 1.3 to 2.6 must go through IP router 0.5 and be screened, even though the shortest physical path between the red and green machines does not go through an IP router. The VLAN architecture has been found to simplify administration, enhance security, and improve the efficiency of campus networks.[22]

A completely different kind of virtual network is often found in multitenant clouds, which may offer to their tenants various services such as load-balancing, packet filtering by firewalls, and application-specific performance enhancements. Such clouds have virtual networks that implement these services by inserting middleboxes into the paths of sessions. In these virtual networks, the major purpose of routing and forwarding is to direct the packets of sessions through middleboxes according to the tenant's service specification.[3,16]

The most unusual networks in this article are named data networks (NDN).[25] In NDN each piece of data has a unique name. For purposes of the networking functions of routing and forwarding, a data server has the name of every piece of data available from it; a server can have many names, and a name can be assigned to many servers. The routing protocol uses advertising and other conventional techniques so that a request for data is usually forwarded to the nearest server with the requested data. In NDN, a session consists of a single request and its response, and there is no source name in the request packet. (A source name would be useless for returning the re-

Figure 3. The architecture of campus network.

In this diagram, all lines between members are bidirectional pairs of links.



[b] The "campus IP network" at the bottom level is a tricky part of the architecture, and will be explained in section entitled The Usage Graph.

sponse to the requestor, because users do not have names and server names are not unique.) Rather, the session protocol leaves traces of the session in every member that forwards the request, so that the response can follow the path in reverse.

NDN is a "future Internet architecture," as mentioned at the outset. In current NDN deployments, wherever NDN links must traverse non-NDN nodes, they are implemented by being layered on top of the public Internet. NDN networks are particularly interesting because their design shows how the session protocol, routing, and forwarding of a network can be highly specialized and tightly integrated.

Composition by Layering

Layering of self-contained networks. The most important operator for the composition of networks is layering, which is simply what happens when one network uses the services of another network, in exactly the sense discussed earlier. More specifically, a link in a user network is implemented by a session in a used network.

The most important operator for the composition of networks is layering, which is when a link in a network is implemented by a session in a used network.

A *usage hierarchy* is a directed acyclic graph whose nodes are networks and whose edges represent composition by link implementation. A *level* in this graph is a set of networks that all have the same graph distance from some reference point. This definition will be refined further.

For example, Figure 3 is derived from a usage hierarchy, with the levels of the graph being represented by vertical placement. The bidirectional link between 2.7 and 2.8 in the campus IP network is implemented by a bidirectional session in the administrators' VLAN that follows the path shown between M7 and M8. The link between M7 and M4 in the VLAN is implemented by a session in the left physical LAN following the spanning-tree path between M7 and M4. Note that machines have distinct members in VLANs and LANs, even though those networks happen to use the same Ethernet design and the same namespace.

Consider the right LAN in Figure 3. It links machines in both the students' and administrators' groups, so it must implement links in at least two VLANs. When the destination of a session in this LAN receives a packet, which member of which VLAN should it deliver the packet to? LAN packets in this architecture have a user-network identifier called a "VLAN tag," which tells the destination which user network is being served by the session.

The shift from a principle of fixed layers to a principle of many self-contained networks encourages a shift in thinking and terminology—from different concepts and terminology for each layer to concepts and terminology that emphasize the similarities among layered networks. Most importantly of all, users of networks—distributed application systems—can be networks themselves, and the distinction between the two concepts weakens.

If the service provided by a session protocol has a specification, then the specified properties of a session are also the guaranteed properties of a link the session implements. For example, the best-known service of IP networks is implemented by the session protocol TCP. A user of TCP sends a stream of bytes, and this byte stream must be received by the user at the other end of the session with no bytes missing or duplicated, and all in the same order in which they were sent.

In the case of TCP the work needed to satisfy this specification is performed by the protocol implementation in the network members at the endpoint machines. IP/TCP packet headers have a *session identifier* (the four-tuple with both names and both ports) and a *user network* (the destination port or "well-known port"). The network has a maximum transmission unit limiting the size of IP packets. So the TCP implementation at the source accepts a byte substream, disassembles it into IP packets, encapsulates each packet in the TCP/IP header, and sends it through the network. When the TCP implementation at the destination receives packets, it decapsulates them by removing the TCP/IP header, requests retransmissions of missing substreams, assembles a complete substream in byte order, and delivers it to the receiver.

Names and directories. Classic descriptions of the Internet associate "domain names" with the application layer, IP "addresses" with the network layer, and MAC addresses with the link layer. In the new model every network simply has a namespace, and network members have names in the namespaces of their networks. In the literature of networking, names in various networks are also referred to as "service names," "identifiers," and "locations."

In every instance of layering composition, a network *A* uses a network *B*. Some members of *A* must be running on the same machines as members of *B*, and interfacing with them to get network services. If *B* must set up sessions dynamically to serve *A*, then there must be a directory mapping names in *A* to the names of the members of *B* on the same machines. For example, a Web request is sent from a client to a server having a domain name in the Web namespace. For an IP network to implement this communication, it must discover the network name (IP address) of the server, which will be the destination of the TCP session carrying the request. DNS is the directory providing this information.[c]

The new model does not constrain internal implementation details of networks. For example, although most networks store member-specific forwarding tables in individual members, in SEATTLE there is a single (although distributed) forwarding table used by all members.[15] And although many networks have centralized directories, in Ethernets the directory information obtained from the Address Resolution Protocol is cached in individual members. Thus forwarding state and directory state cannot always be distinguished by the way they are implemented. But they can always be distinguished by what they are mapping: forwarding state maps destination names to members/names in the same network, while directory state maps names from one network to names in another network.

Service properties and compositional reasoning. A network offers to its users one or more communication services, each specified as a set of properties, and some associated with the use of specific session protocols. Some properties are defined on individual sessions, while others are defined on aggregates of sessions. In general, the properties fall into four categories:

- *Reachability* properties specify which receivers a member can send packets to.
- *Performance* properties specify quantities such as maximum latency, minimum bandwidth, maximum packet loss rate, and faults tolerated.
- *Behavioral* properties are more service-specific. In addition to TCP guarantees, they include synchronization, load balancing among user endpoints, and the requirement that a session must persist despite physical mobility of one or both endpoint machines.
- *Security* properties are diverse. For example, access control is the negation of reachability. Denial-of-service protection supports availability. Security properties on individual sessions include endpoint authentication, data confidentiality, data integrity, and privacy.

In addition to providing specified services, network designers and operators are also concerned with efficient resource allocation, so that the services are provided at minimal cost.

Basic reasoning about composition by layering is easy to explain. There should be a one-to-one mapping between implemented links and implementing sessions. The packet load on the link, possibly fragmented into smaller packets, becomes the packet load on the implementing session. The guaranteed properties of the session become the assumed properties of the implemented link.

Although such rigor is not always needed, it should be possible to reason that a network satisfies its service specifications, and that its use of resources is close to optimal. Network designers have been very successful at this, at least with respect to performance properties. They have learned to abstract the effects of used and using networks, and have developed effective optimization algorithms and tools for self-contained networks.

Reachability, behavioral, and security properties are not so well understood. Next, we discuss examples in which the new model captures the structures and relationships needed for reasoning compositionally about these properties.

Bridging and Security

Bridging. In our model a network has a single administrative authority, which is responsible for providing the network's services with their specified properties. Bridging is a composition operator in which sessions or services are implemented by a set of networks chained end-to-end. With bridging, the two endpoints of a session can be members of different networks. The public Internet consists of a large number of autonomous IP networks, composed by bridging.

There are several variations on bridging, depending on how much structure the bridged networks share. In the simplest case two bridged networks have identical designs and protocols, names of all network members are unique across both networks, and members of both networks have access to the routing and directories of the other. In this simple case, the networks can be bridged by shared links, and little changes except that the reach of both networks is extended. This is how public IP networks are bridged.

In other cases, bridged networks are less similar. They may have different or overlapping namespaces. They may have unshared routing, unshared directories, or other barriers. In these cases a member of one network can still reach a member of a bridged network, but only with the addition of compound sessions. A *compound session* is simply a session in which there is at least one middlebox acting as a *joinbox*. The joinbox serves as a destination for one simple session and a source for another simple session, and maintains state that associates the two simple sessions so it can forward packets from one to the other.[d] If two bridged networks have incompatible session protocols, then a joinbox, acting as a protocol converter, must be the shared element between them.

We will now introduce a simple, familiar example, which will illustrate bridging, trust, and service verification. Figure 4 shows two private networks communicating through the public Internet, although their relationships to the public Internet are

[c] In cases where DNS maps a domain name to the server nearest the client, the domain name does not uniquely identify a server.

[d] A joinbox must change at least one of the source or destination in the session header; it may or may not be a "proxy," which is a session-protocol endpoint. For example, the NAT in Figure 4 is a joinbox and not a proxy.

not symmetric. In this example an employee's laptop using the private IP network in a coffee shop is connected to the public Internet through bridging. At a higher level, using virtual private network (VPN) technology layered on top of the previous networks, the laptop joins the employer's private enterprise network, and accesses a compute server within it. We will look at the bridging first.

It has been a long time since there has been enough room in the IPv4 32-bit namespace to give every networked machine a unique name. Outright exhaustion of the namespace was delayed by the fact that most private networks reuse the same set of private IP addresses. The cost of this strategy is that private IP addresses are ambiguous except in their local context, and a machine with a private address cannot be reached from outside its local network except with a compound session.

In Figure 4, the joinbox for the compound session is the coffee shop's IP router, which incorporates the functionality of network address translation (NAT). The bidirectional compound session is initiated from the private address *X*, to public address **S**. Upon receiving the session-initiating packet, the NAT/router alters it before forwarding, thus making an outgoing session with its own public address **N** as the source. When S accepts this session and sends packets in the reverse direction, it uses reachable **N** as the destination rather than unreachable *X*. In this figure, the dark-gray box represents the public Internet as one network, ignoring the fact it is really a bridging of many networks. Bridging is shown explicitly by the link and session across a network boundary. In the usage hierarchy, the enterprise network uses both lower-level networks.

At the higher level of Figure 4, the enterprise network is also a private IP network, with private addresses *U* and *W*, and public address **S**. The laptop joins the enterprise network by creating a dynamic link to the VPN server. The link is implemented by the IPsec session, so that packets are transmitted in encrypted and authenticated form. The VPN server authenticates the laptop, which has secret credentials issued by the enterprise, and gives it temporary address *U* within the enterprise network. At this point the laptop can initiate a session with compute server *W*, using TCP as the session protocol in the higher-level IP network.

Verification of trustworthy services. To prove security properties, some entities must have responsibilities and be trusted to fulfill them. Normally the entity that is trusted is a machine because the whole machine has a single owner,[e] but trusted to do what, and by whom? A machine can have members of multiple networks, and in each network its member can play a different role.

In networks bridged together in and with the public Internet, as on the lower level of Figure 4, a network's administrative authority owns routers (and other infrastructure machines) and trusts them to behave as specified. Because the administrative authority does not trust the user members (endpoints), the behavior of the routers and other infrastructure machines should be sufficient to provide the specified services in cooperation with well-behaved endpoints, and to protect the network from ill-behaved endpoints. Beyond the technical sources of trust, economic relationships provide incentives for administrative authorities to ensure that networks satisfy their service specifications.[6]

In Figure 4, the employee's laptop and enterprise gateway have network members that are not trusted by their Internet providers, but are trusted by the enterprise. The VPN server does not allow the laptop's member *U* to join the enterprise network until it shows that it is trustworthy by sending secret credentials.

This VPN architecture enforces two security properties:

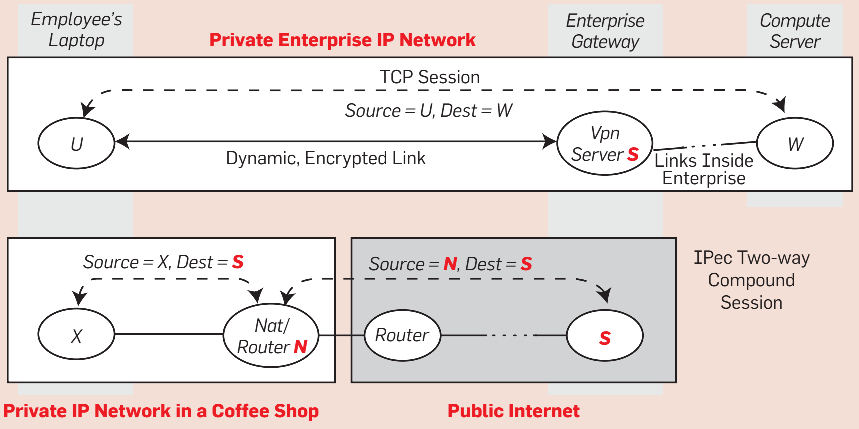
- Only packets originating at members of the enterprise network should be allowed to reach *W*.
- All enterprise data being transmitted outside the walls of the enterprise should have confidentiality and integrity, meaning that no external agent can read or alter the data.

The second property is guaranteed by the IPsec implementation of dynamic links outside enterprise walls. To prove the first property, it is necessary to establish that only packets transmitted on links in the enterprise network (which is not bridged to other networks) are forwarded to *W*. The easiest way to prove this is to rely on the fact that dynamic links of the enterprise network are associated with specific lower-level sessions. Then it is only necessary to check—no matter what packets the public Internet delivers to its member **S**—that the member drops all received packets unless they belong to sessions implementing dynamic links.

The VPN example is especially simple because the security mechanisms at both levels are implemented on the

Figure 4. VPN architecture.

Public names are in boldface red, while private names are not. Light-gray boxes show attachments of members within the same machine.



[e] These terms must be refined slightly to apply to clouds, in which a machine hosts virtual machines.

same machine. The same verification pattern works for more complex security mechanisms, however. The common structures are a secure network layered on top of the public Internet, and a packet-filtering mechanism that prevents harm (including denial-of-service attacks) at the level of the public Internet.[1] The secure overlay carries only approved packets, as enforced by its ingress members. The packet filters are on different machines, and need only have enough knowledge to reject packets not belonging to sessions implementing links of the overlay.

These examples barely scratch the surface of network security. Nevertheless, a broad survey of security mechanisms[24] has shown that the compositional model is important for understanding all aspects of security, and for working toward a comprehensive proof framework. The model is especially valuable for discovering how security interacts with other aspects of network architecture such as session protocols, routing, virtualization, and middleboxes.

The Usage Graph

One of the most interesting aspects of composition is that sometimes the "usage hierarchy" is a convenient fiction, because composition creates a usage graph with cycles. It is still useful to think in terms of usage hierarchies, provided that we remember they are approximate abstractions with localized exceptions.

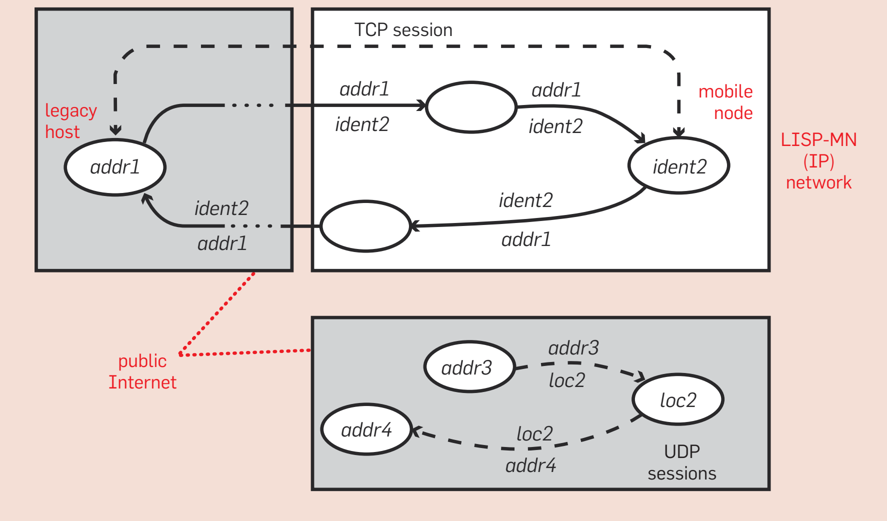
Mobility is a network service that preserves reachability to a network member, and may even preserve the member's ongoing sessions, even though the member's machine is moving. One kind of mobility is provided by LISP Mobile Node[8,9] (for a survey of all kinds of mobility, see Zave and Rexford[23]). With LISP-MN, a machine has a network member with a persistent IP address called an "identifier." In a lower-level IP network, the machine has a member with a temporary, location-dependent IP address called a "location." As a new and lightweight way to provide mobility, LISP-MN must interoperate with the public Internet. Figure 5 shows how. As in Figure 4, the public Internet is depicted as if it were one network.

At the top level of this figure, the public Internet is bridged with a LISP-MN network, which is a specialized IP network. The LISP-MN network owns a range of IP addresses, from which identifiers are drawn. Because of the bridging, a legacy host with IP address *addr1* has been able to initiate a TCP session with a mobile node whose identifier is *ident2*.

The shared elements for bridging are the unlabeled middleboxes. In both networks these middleboxes resemble IP routers, in that they forward packets and do not behave as session endpoints. The middleboxes advertise the mobile range of IP addresses into the public Internet, which means each packet destined for an address in this range will be forwarded to one of them. The LISP-MN network has a directory mapping identifiers of mobile nodes to their current locations. When such a middlebox receives its first packet for *ident2* (or first in a long time), it gets *ident2*'s location *loc2* from the directory, creates a dynamic link to *ident2*, and forwards the packet on it. Subsequent packets to *ident2* use the same link.

Figures 5. The interoperation of LISP-MN with the public Internet.

Each link (solid line), session (dashed line), or path of links and forwarders (solid line broken with dots) is labeled above with the source of the packets traveling on it, and below with their destinations.



The LISP-MN network is layered on top of the public Internet, so that dynamic LISP links are implemented by public UDP sessions. On the same machines as the three members of the LISP-MN network there are members of the public Internet with IP addresses *addr3*, *addr4*, and *loc2*, and these are the endpoints of the UDP sessions. When a mobile node changes its location, it notifies all the middleboxes with which it has dynamic links, and also updates the directory. The UDP sessions will move to the new location, but the LISP-MN links will remain.

Like Figures 3 and 4, Figure 5 uses vertical position to imply a usage graph. In this usage graph, the LISP-MN network is both bridged with the public Internet (at the same level) and layered on it. To avoid drawing the cycle, we depict the public Internet in two places. This graph shows a common pattern for interoperation of special-purpose IP networks with the public Internet.

Figure 3 is another example of a usage graph with a cycle in it. As in Figure 5, rather than drawing a cycle, we have put a network— here the campus IP network—in the figure twice. At the bottom level of the figure, the only physical connection between LANs is the campus IP network. The link shown is exactly the same as the link between 0.4 and 0.5 at the top level of the figure. When an IP packet is sent from source 2.7 to destination 2.8, it is encapsulated in a VLAN header with source M7 and destination M8. When that packet is traversing the VLAN link between M4 and M5, it is further encapsulated in an IP header with source 0.4 and destination 0.5.

This packet format is called the "VXLAN" format.

Special-purpose virtual links in IP networks are often called "tunnels." Our model provides a structured view of tunnels, clarifying the roles of network members at the upper and lower levels of tunnel endpoints, the state that each network member requires, and the fields that must be present in packet headers. This uniformity across levels can explain confusing designs and make them analyzable. For example, even though a network can use itself in a usage graph, a network link must never use itself.

Potential Benefits of the Compositional Model

Since 1993, the Internet has evolved by means of new networks and new compositions. The Internet today is a vast collection of networks comprised in a rich variety of ways by layering and bridging, including being composed with themselves. Networks are easy to add locally (campus networks, cloud computing) or at high levels of the approximate usage hierarchy (mobility, distributed systems). They are slower to disseminate when both global and low in the hierarchy (IPv6).

This evolution, while necessary to keep up with increased demand, new technology, and many new requirements, has created tremendous complexity. First and foremost, our compositional model describes the current complex Internet as precisely as the classic Internet architecture described the Internet of 1993. Because it is inherently modular, it also has the potential to organize, explain, and simplify as well as to describe.

Based on our experience applying the model to many kinds of networks and aspects of networking, there are two primary reasons for adding a new network to the global Internet architecture:

- The network provides a specialized service or unusual cost/performance trade-off through mechanisms that are not compatible with the general-purpose classic Internet design.
- There is a need for two different instances of a network structure with two different purposes. As in LISP-MN, member names might be either permanent identifiers or temporary locations. For another example, the topology of a network might be dictated by security partitions (VLANs) or by paths through required middleboxes, as well as by physical connectivity.

Layered networks hide information, which can make problem diagnosis very difficult.[19] On the other hand, separation of concerns into different networks is a way of taming complexity. This is especially obvious when networks are being added for the second reason, and distinct topologies (for example) are maintained by distinct networks. Also, very often, it is more efficient to compose two networks than to intertwine distinct structures in the same network. This is illustrated well by Qazi et al.,[16] which shows that the conflation of a middlebox topology and a physical topology would cause a combinatorial explosion of router state.

The most immediate potential benefits of the new model are based on its capacity to explain the complexity that is already present and must be dealt with. The model can be formalized through analytic tools and reasoning technology, in support of robustness and verification of trustworthy services. We also believe the model should be used in graduate-level teaching, to cover a wider variety of networks in a shorter period of time, and to encourage recognition of patterns and principles.

Next, the model has the potential to improve current design and development of software-defined networks. Reusable patterns would both increase the availability of different points in a trade-off space, and make each easier to deploy by means of reusable or generated software. Optimizations should become easier to apply, because the model can help us reason that they are safe.

Finally, a compositional model may help us to find a simpler future Internet architecture that truly meets foreseeable requirements and might even adapt to unforeseeable ones. Perhaps, with study of compositional principles and compositional reasoning, we can discover optimal uses of composition, in configurations that exploit its benefits and ameliorate its disadvantages. This could be the basis of network architectures that offer both flexibility and manageability. Pushing Internet evolution in this direction would be a truly worthy goal.

References

1. Andersen, D.G. Mayday: Distributed filtering for Internet services. In *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems*, 2013.
2. Andersen, D.G. Balakrishnan, H., Feamster, N., Koponen, T., Moon, D. and Shenker, S. Accountable Internet Protocol (AIP). In *Proceedings of ACM SIGCOMM*, 2008.
3. Benson, T., Akella, A., Shaikh, A. and Sahu, S. Clous-NaaS: A cloud networking platform for enterprise applications. In *Proceedings of the 2nd ACM Symposium on Cloud Computing*, 2011.
4. Blumenthal, M.S. and Clark, D.G. Rethinking the design of the Internet: The end-to-end arguments vs. the brave new world. *ACM Trans. Internet Technology 1*, 1 (Aug. 2001), 70—109.
5. Clark, D.D. The design philosophy of the DARPA Internet protocols. In *Proceedings of ACM SIGCOMM*, 1988. ACM.
6. Clark, D.D., Wroclawski, J., Sollins, K.R. and Braden, R. Tussle in cyberspace: Defining tomorrow's Internet. *IEEE/ACM Trans. Networking 13*, 3 (June 2005), 462–475.
7. Day, J. *Patterns in Network Architecture: A Return to Fundamentals*. Prentice Hall, 2008.
8. Farinacci, D., Fuller, V., Meyer, D. and Lewis, D. The Locator/ID Separation Protocol (LISP). IETF Request for Comments 6830. (Jan. 2013).
9. Farinacci, D., Lewis, D., Meyer, D., and White, C. 2017. LISP Mobile Node. IETF Network Working Group Internet Draft draft-ietf-lisp-mn-04. (Oct. 2017).
10. Feamster, N., Rexford, J. and Zegura, E. The road to SDN: An intellectual history of programmable networks. *ACM Queue; https://queue.acm.org/detail.cfm?id=2560327*.
11. Fuller, V., Li, T., Yu, J. and Varadhan, K. Classless inter-domain routing (CIDR): An address assignment and aggregation strategy. IETF Network Working Group Request for Comments 1519. (1993).
12. Handley, M. Why the Internet only just works. *BT Technology Journal 24*, 3 (July 2006), 119–129.
13. ITU. Information Technology—Open Systems Interconnection—Basic Reference Model: The basic model. ITU-T Recommendation X.200. (1994).
14. Karsten, M., Keshav, S. and Prasad, S. An axiomatic basis for communication. In *Proceedings of HotNets-V*, 2006. ACM.
15. Kim, C., Caesar, M. and Rexford, J. SEATTLE: A scalable Ethernet architecture for large enterprises. *ACM Trans. Computer Systems 29*, 1 (2011).
16. Qazi, Z.A., Tu, C.C., Chiang, L., Miao, R., Sekar, V. and Yu, M. SIMPLE-fying middlebox policy enforcement using SDN. In *Proceedings of ACM SIGCOMM*. 2013.
17. Roscoe, T. The end of Internet architecture. In *Proceedings of the 5th Workshop on Hot Topics in Networks*, 2006.
18. Saltzer, J., Reed, D. and Clark, D.D. End-to-end arguments in system design. *ACM Trans. Computer Systems 2*, 4 (Nov. 1984), 277–288.
19. Spatscheck, O. Layers of success. *IEEE Internet Computing 17*, 1 (2013), 3–6.
20. Venkataramani, A., Kurose, J.F., Raychaudhuri, D., Nagaraja, K., Banerjee, S. and Mao, Z.M. MobilityFirst: A mobility-centric and trustworthy Internet architecture. *ACM SIGCOMM Computer Communication Review 44*, 3 (July 2014), 74–80.
21. Wang, Y., Matta, I., Esposito, F. and Day, J. Introducing ProtoRINA: A prototype for programming recursive-networking policies. *ACM SIGCOMM Computer Communications Review 44*, 3 (July 2014).
22. Yu, M., Rexford, J., Sun, X., Rao, S. and Feamster, N. A survey of virtual LAN usage in campus networks. *IEEE Communications 49*, 7 (July 2011), 98–103.
23. Zave, P. and Rexford, J. The design space of network mobility. In Recent Advances in Networking, Olivier Bonaventure and Hamed Haddadi (Eds). *ACM SIGCOMM*, 2013.
24. Zave, P. and Rexford, J. Network Security; https://www.cs.princeton.edu/courses/archive/fall18/cos561/papers/Security18.pdf.
25. Zhang, L., Afanasyev, A., Burke, J. and Jacobson, V. Named Data Networking. *ACM SIGCOMM Computer Communication Review 44*, 3 (July 2014), 66–73.

Pamela Zave (Pamela@pamelazave.com) is a researcher in the Department of Computer Science at Princeton University, Princeton, NJ, USA.

Jennifer Rexford (jrex@cs.princeton.edu) is the Gordon Y.S. Wu Professor of Engineering in the Department of Computer Science at Princeton University, Princeton, NJ, USA.

DOI:10.1145/3232535

The need for deeply understanding when algorithms work (or not) has never been greater.

BY TIM ROUGHGARDEN

Beyond Worst-Case Analysis

COMPARING DIFFERENT ALGORITHMS is hard. For almost any pair of algorithms and measure of algorithm performance like running time or solution quality, each algorithm will perform better than the other on some inputs.[a] For example, the insertion sort algorithm is faster than merge sort on already-sorted arrays but slower on many other inputs. When two algorithms have incomparable performance, how can we deem one of them "better than" the other?

Worst-case analysis is a specific modeling choice in the analysis of algorithms, where the overall performance of an algorithm is summarized by its worst performance on any input of a given size. The "better" algorithm is then the one with superior worst-case performance. Merge sort, with its worst-case asymptotic running time of $\Theta(n \log n)$ for arrays of length n, is better in this sense than insertion sort, which has a worst-case running time of $\Theta(n^2)$.

While crude, worst-case analysis can be tremendously useful, and it is the dominant paradigm for algorithm analysis in theoretical computer science. A good worst-case guarantee is the best-case scenario for an algorithm, certifying its general-purpose utility and absolving its users from understanding which inputs are relevant to their applications. Remarkably, for many fundamental computational problems, there are algorithms with excellent worst-case performance guarantees. The lion's share of an undergraduate algorithms course comprises algorithms that run in linear or near-linear time in the worst case.

For many problems a bit beyond the scope of an undergraduate course, however, the downside of worst-case analysis rears its ugly head. Here, I review three classical examples where worst-case analysis gives misleading or useless advice about how to solve a problem; further examples in modern machine learning are described later. These examples motivate the alternatives to worst-case analysis described in the article.[b]

The simplex method for linear programming. Perhaps the most famous failure of worst-case analysis concerns linear programming, the problem of optimizing a linear func-

» key insights

- **Worse-case analysis takes a "Murphy's Law" approach to algorithm analysis, which is too crude to give meaningful algorithmic guidance for many important problems, including linear programming, clustering, caching, and neural network training.**
- **Research going "beyond worst-case analysis" articulates properties of realistic inputs, and proves rigorous and meaningful algorithmic guarantees for inputs with these properties.**
- **Much of the present and future research in the area is motivated by the unreasonable effectiveness of machine learning algorithms.**

[a] In rare cases a problem admits an instance-optimal algorithm, which is as good as every other algorithm on every input, up to a constant factor.[23] For most problems, there is no instance-optimal algorithm, and there is no escaping the incomparability of different algorithms.

[b] For many more examples, analysis frameworks, and applications, see the author's lecture notes.[36]

COLLAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

tion subject to linear constraints (Figure 1). Dantzig's simplex method is an algorithm from the 1940s that solves linear programs using greedy local search on the vertices on the solution set boundary, and variants of it remain in wide use to this day. The enduring appeal of the simplex method stems from its consistently superb performance in practice. Its running time typically scales modestly with the input size, and it routinely solves linear programs with millions of decision variables and constraints. This robust empirical performance suggested the simplex method might well solve every linear program in a polynomial amount of time.

In 1972, Klee and Minty showed by example that there are contrived linear programs that force the simplex method to run in time exponential in the number of decision variables (for all of the common "pivot rules" for choosing the next vertex). This illustrates the first potential pitfall of worst-case analysis: overly pessimistic performance predictions that cannot be taken at face value. The running time of the simplex method is polynomial for all practical purposes, despite the exponential prediction of worst-case analysis.

To add insult to injury, the first worst-case polynomial-time algorithm for linear programming, the ellipsoid method, is not competitive with the simplex method in practice.[c]

[c] Interior-point methods, developed five years later, lead to algorithms that both run in worst-case polynomial time and are competitive with the simplex method in practice.

Figure 1. A two-dimensional linear programming problem.

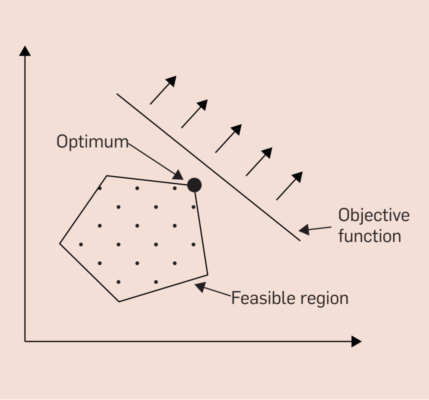
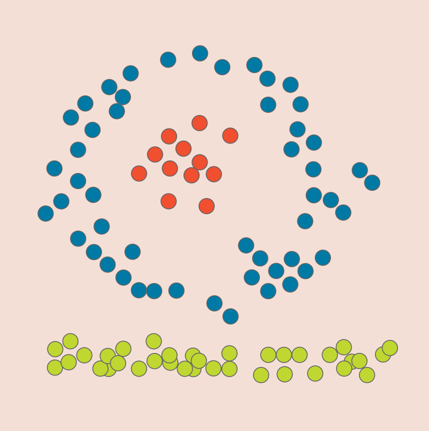


Figure 2. One possible way to group data points into three clusters.



Taken at face value, worst-case analysis recommends the ellipsoid method over the empirically superior simplex method. One framework for narrowing the gap between these theoretical predictions and empirical observations is *smoothed analysis*, discussed later in this article.

Clustering and *NP*-hard optimization problems. Clustering is a form of unsupervised learning (finding patterns in unlabeled data), where the informal goal is to partition a set of points into "coherent groups" (Figure 2). One popular way to coax this goal into a well-defined computational problem is to posit a numerical objective function over clusterings of the point set, and then seek the clustering with the best objective function value. For example, the goal could be to choose k cluster centers to minimize the sum of the distances between points and their nearest centers (the k-median objective) or the sum of the squared such distances (the k-means objective). Almost all natural optimization problems that are defined over clusterings are *NP*-hard.

In practice, clustering is not viewed as a particularly difficult problem. Lightweight clustering algorithms, like Lloyd's algorithm for k-means and its variants, regularly return the intuitively "correct" clusterings of real-world point sets. How can we reconcile the worst-case intractability of clustering problems with the empirical success of relatively simple algorithms?[d]

One possible explanation is that *clustering is hard only when it doesn't matter*.[18] For example, if the difficult instances of an *NP*-hard clustering problem look like a bunch of random unstructured points, who cares? The common use case for a clustering algorithm is for points that represent images, or documents, or proteins, or some other objects where a "meaningful clustering" is likely to exist. Could instances with a meaningful clustering be easier than worst-case instances? This article surveys recent theoretical developments that support an affirmative answer.

Cache replacement policies. Consider a system with a small fast memory (the cache) and a big slow memory. Data is organized into blocks called *pages*, with up to k different pages fitting in the cache at once. A page request results in either a cache hit (if the page is already in the cache) or a cache miss (if not). On a cache miss, the requested page must be brought into the cache. If the cache is already full, then some page in it must be evicted. A cache policy is an algorithm for making these eviction decisions. Any systems textbook will recommend aspiring to the least recently used (LRU) policy, which evicts the page whose most recent reference is furthest in the past. The same textbook will explain why: real-world page request sequences tend to exhibit locality of reference, meaning that recently requested pages are likely to be requested again soon. The LRU policy uses the recent past as a prediction for the near future. Empirically, it typically suffers fewer cache misses than competing policies like first-in first-out (FIFO).

Sleator and Tarjan[37] founded the area of *online algorithms*, which are algorithms that must process their input as it arrives over time (like cache policies). One of their first observations was that worst-case analysis, straightforwardly applied, provides no useful insights about the performance of different cache replacement policies. For every deterministic policy and cache size k, there is a pathological page request sequence that triggers a page fault rate of 100%, even though the optimal clairvoyant replacement policy (known as Bélády's algorithm) would have a page fault rate of at most $(1/k)$%. This observation is troublesome both for its absurdly pessimistic performance prediction and for its failure to differentiate between competing replacement policies (like LRU vs. FIFO). One solution, discussed next, is to choose an appropriately fine-grained parameterization of the input space and to assess and compare algorithms using parameterized guarantees.

Models of Typical Instances

Maybe we shouldn't be surprised that worst-case analysis fails to advocate LRU over FIFO. The empirical superiority of LRU is due to the special structure in real-world page request sequences—locality of reference—and traditional worst-case analysis provides no vocabulary to speak about this structure.[e] This is what work on "beyond worst-case analysis" is all about: *articulating properties of "real-world" inputs, and proving rigorous and meaningful algorithmic guarantees for inputs with these properties.*

Research in the area has both a scientific dimension, where the goal is to develop transparent mathemati-

[d] More generally, optimization problems are more likely to be NP-hard than not. In many cases, even computing an approximately optimal solution is an NP-hard problem (see Trevisan[36] for example). Whenever an efficient algorithm for such a problem performs better on real-world instances than (worst-case) complexity theory would suggest, there's an opportunity for a refined and more accurate theoretical analysis.

[e] If worst-case analysis has an implicit model of data, then it's the "Murphy's Law" data model, where the instance to be solved is an adversarially selected function of the chosen algorithm. Outside of cryptographic applications, this is a rather paranoid and incoherent way to think about a computational problem.

cal models that explain empirically observed phenomena about algorithm performance, and an engineering dimension, where the goals are to provide accurate guidance about which algorithm to use for a problem and to design new algorithms that perform particularly well on the relevant inputs.

One exemplary result in beyond worst-case analysis is due to Albers et al.,[2] for the online paging problem described in the introduction. The key idea is to parameterize page request sequences according to how much locality of reference they exhibit, and then prove parameterized worst-case guarantees. Refining worst-case analysis in this way leads to dramatically more informative results.[f]

Locality of reference is quantified via the size of the working set of a page request sequence. Formally, for a function $f : \mathbb{N} \to \mathbb{N}$, we say that a request sequence *conforms to f* if, in every window of w consecutive page requests, at most $f(w)$ distinct pages are requested. For example, the identity function $f(w) = w$ imposes no restrictions on the page request sequence. A sequence can only conform to a sublinear function like $f(w) = \lceil \sqrt{w} \rceil$ or $f(w) = \lceil 1 + \log_2 w \rceil$ if it exhibits locality of reference.[g]

The following worst-case guarantee is parameterized by a number $\alpha_f(k)$, between 0 and 1, that we discuss shortly; recall that k denotes the cache size. It assumes that the function f is "concave" in the sense that the number of inputs with value x under f (that is, $|f^{-1}(x)|$) is nondecreasing in x.

Theorem 1 (Albers et al.[2])

(a) For every f and k and every deterministic cache replacement policy, the worst-case page fault rate (over sequences that conform to f) is at least $\alpha_f(k)$.

(a) For every f and k and every sequence that conforms to f, the page fault rate of the LRU policy is at most $\alpha_f(k)$.

(b) There exists a choice of f and k, and a page request sequence that conforms to f, such that the page fault rate of the FIFO policy is strictly larger than $\alpha_f(k)$.

Parts (a) and (*b*) prove the worst-case optimality of the LRU policy in a strong sense, f-by-f and k-by-k. Part (c) differentiates LRU from FIFO, as the latter is suboptimal for some (in fact, many) choices of f and k.

The guarantees in Theorem 1 are so good that they are meaningful even when taken at face value—for sublinear f's, $\alpha_f(k)$ goes to 0 reasonably quickly with k. For example, if $f(w) = \lceil \sqrt{w} \rceil$, then $\alpha_f(k)$ scales with $1/\sqrt{k}$. Thus, with a cache size of 10,000, the page fault rate is always at most 1%. If $f(w) = \lceil 1 + \log_2 w \rceil$, then $\alpha_f(k)$ goes to 0 even faster with k, roughly as $k/2^k$.[h]

Stable Instances

Are point sets with meaningful clusterings easier to cluster than worst-case point sets? Here, we describe one way to define a "meaningful clustering," due to Bilu and Linial;[12] for others, see Ackerman and Ben-David,[1] Balcan et al.,[9] Daniely et al.,[18] Kumar and Kannan,[29] and Ostrovsky et al.[34]

The maximum cut problem. Suppose you have a bunch of data points representing images of cats and images of dogs, and you would like to automatically discover these two groups. One approach is to reduce this task to the *maximum cut* problem, where the goal is to partition the vertices V of a graph G with edges E and nonnegative edge weights into two groups, while maximizing the total weight of the edges that have one endpoint in each group. The reduction forms a complete graph G, with vertices corresponding to the data points, and assigns a weight w_e to each edge e indicating how dissimilar its endpoints are. The maximum cut of G is a 2-clustering that tends to put dissimilar pairs of points in different clusters.

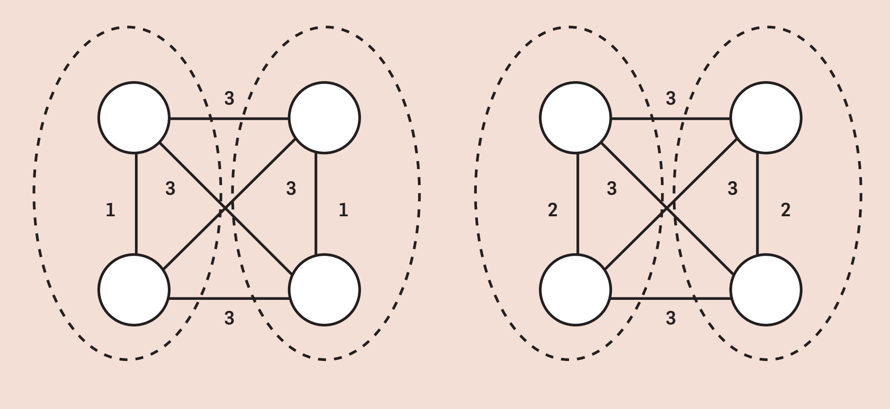
There are many ways to quantify "dissimilarity" between images, and different definitions might give different optimal 2-clusterings of the data points. One would hope that, for a range of reasonable measures of dissimilarity, the maximum cut in the example above would have all cats on one side and all dogs on the other. In other words, the maximum cut should be invariant under minor changes to the specification of the edge weights (Figure 3).

Definition 2 (Bilu and Linial[12]). An instance $G = (V, E, w)$ of the maximum cut problem is *γ-perturbation stable* if, for all ways of multiplying the weight w_e of each edge e by a factor $a_e \in [1, \gamma]$, the optimal solution remains the same.

A perturbation-stable instance has a "clearly optimal" solution—a uniqueness assumption on steroids—thus formalizing the idea of a "meaningful clustering." In machine learning parlance, perturbation stability can be viewed as a type of "large margin" assumption.

The maximum cut problem is *NP*-hard in general. But what about the special case of γ-perturbation-stable in-

Figure 3. In a perturbation-stable maximum cut instance, the optimal solution is invariant under small perturbations to the edges' weights.



[f] Parameterized guarantees are common in the analysis of algorithms. For example, the field of parameterized algorithms and complexity has developed a rich theory around parameterized running time bounds (see the book by Cygan et al.[16]). Theorem 1 employs an unusually fine-grained and problem-specific parameterization, and in exchange obtains unusually accurate and meaningful results.

[g] The notation $\lceil x \rceil$ means the number x, rounded up to the nearest integer.

[h] See Albers et al.[2] for the precise closed-form formula for $\alpha_f(k)$ in general.

stances? As γ increases, fewer and fewer instances qualify as γ-perturbation stable. Is there a sharp stability threshold—a value of γ where the maximum cut problem switches from *NP*-hard to polynomial-time solvable?

Makarychev et al.[30] largely resolved this question. On the positive side, they showed that if γ is at least a slowly growing function of the number of vertices n, then the maximum cut problem can be solved in polynomial time for all γ-perturbation stable instances.[i] Makarychev et al. use techniques from the field of metric embeddings to show that, in such instances, the unique optimal solution of a certain semidefinite programming relaxation corresponds precisely to the maximum cut.[j] Semidefinite programs are convex programs, and can be solved to arbitrary precision in polynomial time. There is also evidence that the maximum cut cannot be recovered in polynomial time in γ-perturbation-stable instances for much smaller values of γ.[30]

Other clustering problems. Bilu and Linial[12] defined γ-perturbation-stable instances specifically for the maximum cut problem, but the definition makes sense more generally for any optimization problem with a linear objective function. The study of γ-perturbation-stable instances has been particularly fruitful for *NP*-hard clustering problems in metric spaces, where interpoint distances are required to satisfy the triangle inequality. Many such problems, including the k-means, k-median, and k-center problems, are polynomial-time solvable already in 2-perturbation-stable instances.[5,10] The algorithm in Angelidakis et al.,[5] like its precursor in Awasthi et al.,[8] is inspired by the well known single-linkage clustering algorithm. It computes a minimum spanning tree (where edge weights are the interpoint distances) and uses dynamic programming to optimally remove k - 1 edges to define k clusters. To the extent that we are comfortable identifying "instances with a meaningful clustering" with 2-perturbation-stable instances, these results give a precise sense in which clustering is hard only when it doesn't matter.[k]

The unreasonable effectiveness of modern machine learning algorithms has thrown down the gauntlet to algorithms researchers, and there is perhaps no other problem domain with a more urgent need for the beyond worst-case approach.

Overcoming *NP*-hardness. Polynomial-time algorithms for γ-perturbation-stable instances continue the age-old tradition of identifying "islands of tractability," meaning polynomial-time solvable special cases of *NP*-hard problems. Two aspects of these results diverge from a majority of 20th century research on tractable special cases. First, perturbation-stability is not an easy condition to check, in contrast to a restriction like graph planarity or Horn-satisfiability. Instead, the assumption is justified with a plausible narrative about why "real-world instances" might satisfy it, at least approximately. Second, in most work going beyond worst-case analysis, the goal is to study general-purpose algorithms, which are well defined on all inputs, and use the assumed instance structure only in the algorithm analysis (and not explicitly in its design). The hope is the algorithm continues to perform well on many instances not covered by its formal guarantee. The results here for mathematical programming relaxations and single-linkage-based algorithms are good examples of this paradigm.

Analogy with sparse recovery. There are compelling parallels between the recent research on clustering in stable instances and slightly older results in a field of applied mathematics known as *sparse recovery*, where the goal is to reverse engineer a "sparse" object from a small number of clues about it. A common theme in both areas is identifying relatively weak conditions under which a tractable mathematical programming relaxation of an *NP*-hard problem is guaranteed to be exact, meaning the original problem and its relaxation have the same optimal solution.

For example, a canonical problem in sparse recovery is *compressive sensing*, where the goal is to recover

[i] Specifically, $\gamma = \Omega(\sqrt{\log n} \log \log n)$.

[j] In general, the optimal solution of a linear or semidefinite programming relaxation of an *NP*-hard problem is a "fractional solution" that does not correspond to a feasible solution to the original problem.

[k] A relaxed and more realistic version of perturbation-stability allows small perturbations to make small changes to the optimal solution. Many of the results mentioned in this section can be extended to instances meeting this relaxed condition, with a polynomial-time algorithm guaranteed to recover a solution that closely resembles the optimal one.[5,9,30]

an unknown sparse signal (a vector of length n) from a small number m of linear measurements of it. Equivalently, given an m x n measurement matrix A with $m \ll n$ and the measurement results $b = Az$, the problem is to figure out the signal z. This problem has several important applications, for example in medical imaging. If z can be arbitrary, then the problem is hopeless: since $m < n$, the linear system $Ax = b$ is underdetermined and has an infinite number of solutions (of which z is only one). But many real-world signals are (approximately) k-sparse in a suitable basis for small k, meaning that (almost) all of the mass is concentrated on k coordinates.[1] The main results in compressive sensing show that, under appropriate assumptions on A, the problem can be solved efficiently even when m is only modestly bigger than k (and much smaller than n).[15,20] One way to prove these results is to formulate a linear programming relaxation of the (NP-hard) problem of computing the sparsest solution to $Ax = b$, and then show this relaxation is exact.

Planted and Semi-Random Models

Our next genre of models is also inspired by the idea that interesting instances of a problem should have "clearly optimal" solutions, but differs from the stability conditions in assuming a generative model—a specific distribution over inputs. The goal is to design an algorithm that, with high probability over the assumed input distribution, computes an optimal solution in polynomial time.

The planted clique problem. In the *maximum clique* problem, the input is an undirected graph $G = (V, E)$, and the goal is to identify the largest subset of vertices that are mutually adjacent. This problem is NP-hard, even to approximate by any reasonable factor. Is it easy when there is a particularly prominent clique to be found?

Jerrum[27] suggested the following generative model: There is a fixed set V of n vertices. First, each possible edge (u, v) is included independently with 50% probability. This is also known as an Erdös-Renyi random graph with edge density ½. Second, for a parameter $k \in \{1, 2, \ldots, n\}$, a subset $Q \subseteq V$ of k vertices is chosen uniformly at random, and all remaining edges with both endpoints in Q are added to the graph (thus making Q a k-clique).

How big does k need to be before Q becomes visible to a polynomial-time algorithm? The state of the art is a spectral algorithm of Alon et al.,[3] which recovers the planted clique Q with high probability provided k is at least a constant times $\sqrt{n}$. Recent work suggests that efficient algorithms cannot recover Q for significantly smaller values of k.[11]

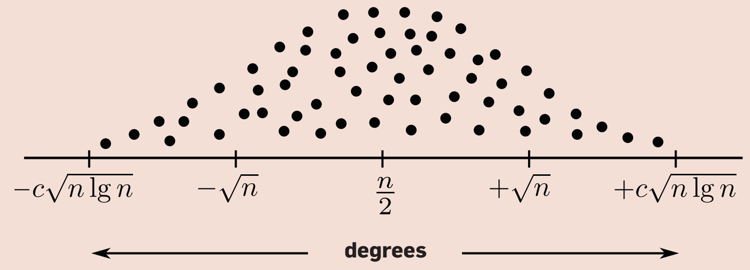
An unsatisfying algorithm. The algorithm of Alon et al.[3] is theoretically interesting and plausibly useful. But if we take k to be just a bit bigger, at least a constant times $\sqrt{n \log n}$, then there is an uninteresting and useless algorithm that recovers the planted clique with high probability: return the k vertices with the largest degrees. To see why this algorithm works, think first about the sampled Erdös-Renyi random graph, before the clique Q is planted. The expected degree of each vertex is $\approx n/2$, with standard deviation $\approx \sqrt{n}/2$. Textbook large deviation inequalities show that, with high probability, the degree of every vertex is within $\approx \sqrt{\ln n}$ standard deviations of its expectation (Figure 4). Planting a clique Q of size $a\sqrt{n \log n}$, for a sufficiently large constant a, then boosts the degrees of all of the clique vertices enough that they catapult past the degrees of all of the non-clique vertices.

What went wrong? The same thing that often goes wrong with pure average-case analysis—the solution is brittle and overly tailored to a specific distributional assumption. How can we change the input model to encourage the design of algorithms with more robust guarantees? Can we find a sweet spot between average-case and worst-case analysis?

Semi-random models. Blum and Spencer[13] proposed studying *semi-random* models, where nature and an adversary collaborate to produce an input. In many such models, nature first samples an input from a specific distribution (like the probabilistic planted clique model noted here), which is then modified by the adversary before being presented as an input to an algorithm. It is important to restrict the adversary's power, so that it cannot simply throw out nature's starting point and replace it with a worst-case instance. Feige and Killian[24] suggested studying *monotone adversaries*, which can only modify the input by making the optimal solution "more obviously optimal." For example, in the semi-random version of the planted clique problem, a monotone adversary is only allowed to remove edges that are not in the planted clique Q—it cannot remove edges from Q or add edges outside Q.

Semi-random models with a monotone adversary may initially seem no harder than the planted models that they generalize. But let's return to the planted clique model with $k = \Omega(\sqrt{n \log n}$), where the "top-k degrees" algorithm succeeds with high probability when there is no adversary. A monotone adversary can easily foil this algorithm in the semi-random planted clique model, by removing edges between clique

Figure 4. Degree distribution of an Erdős–Rényi graph with edge density ½, before planting the *k*-clique *Q*. If *k* = Ω ($\sqrt{(n \lg n}$), then the planted clique will consist of the *k* vertices with the highest degrees.



[1] For example, audio signals are typically approximately sparse in the Fourier basis, images in the wavelet basis.

and non-clique vertices to decrease the degrees of the former back down to ≈ $n/2$. Thus the semi-random model forces us to develop smarter, more robust algorithms.[m]

For the semi-random planted clique model, Feige and Krauthgamer[24] gave a polynomial-time algorithm that recovers the clique with high probability provided $k = \Omega(\sqrt{n})$. The spectral algorithm by Alon et al.[3] achieved this guarantee only in the standard planted clique model, and it does not provide any strong guarantees for the semi-random model. The algorithm of Feige and Krauthgamer[24] instead uses a semidefinite programming relaxation of the problem. Their analysis shows that this relaxation is exact with high probability in the standard planted clique model (provided $k = \Omega(\sqrt{n})$), and uses the monotonicity properties of optimal mathematical programming solutions to argue this exactness cannot be sabotaged by any monotone adversary.

Smoothed Analysis

Smoothed analysis is another example of a semi-random model, now with the order of operations reversed: an adversary goes first and chooses an arbitrary input, which is then perturbed slightly by nature. Smoothed analysis can be applied to any problem where "small perturbations" make sense, including most problems with real-valued inputs. It can be applied to any measure of algorithm performance, but has proven most effective for running time analyses.

Like other semi-random models, smoothed analysis has the benefit of potentially escaping worst-case inputs (especially if they are "isolated"), while avoiding overfitting a solution to a specific distributional assumption. There is also a plausible narrative about why real-world inputs are captured by this framework: whatever problem you would like to solve, there are inevitable inaccuracies in its formulation (from measurement error, uncertainty, and so on).

The simplex method. Spielman and Teng[38] developed the smoothed analysis framework with the specific goal of proving that bad inputs for the simplex method are exceedingly rare. Average case analyses of the simplex method from the 1980s (for example, Borgwardt[14]) provide evidence for this thesis, but smoothed analysis provides more robust support for it.

The perturbation model in Spielman and Teng[38] is: independently for each entry of the constraint matrix and right-hand side of the linear program, add a Gaussian (that is, normal) random variable with mean 0 and standard deviation σ.[n] The parameter σ interpolates between worst-case analysis (when $\sigma = 0$) and pure average-case analysis (as $\sigma \to \infty$, the perturbation drowns out the original linear program). The main result states that the expected running time of the simplex method is polynomial as long as typical perturbations have magnitude at least an inverse polynomial function of the input size (which is small!).

Theorem 3 (Spielman and Teng[38])

For every initial linear program, in expectation over the perturbation to the program, the running time of the simplex method is polynomial in the input size and in $^1/\sigma$.

The running time blow-up as $\sigma \to 0$ is necessary because the worst-case running time of the simplex method is exponential. Several researchers have devised simpler analyses and better polynomial running times, most recently Dadush and Huiberts.[17] All of these analyses are for a specific pivot rule, the "shadow pivot rule." The idea is to project the high-dimensional feasible region of a linear program onto a plane (the "shadow") and run the simplex method there. The hard part of proving Theorem 3 is showing that, with high probability over nature's perturbations, the perturbed instance is well-conditioned in the sense that each step of the simplex method makes significant progress traversing the boundary of the shadow.

Local search. A local search algorithm for an optimization problem maintains a feasible solution, and iteratively improves that solution via "local moves" for as long as possible, terminating with a locally optimal solution. Local search heuristics are ubiquitous in practice, in many different application domains. Many such heuristics have an exponential worst-case running time, despite always terminating quickly in practice (typically within a sub-quadratic number of iterations). Resolving this disparity is right in the wheelhouse of smoothed analysis. For example, Lloyd's algorithm for the k-means problem can require an exponential number of iterations to converge in the worst case, but needs only an expected polynomial number of iterations in the smoothed case (see Arthur et al.[7] and the references therein).[o]

Much remains to be done, however. For a concrete challenge problem, let's revisit the maximum cut problem. The input is an undirected graph $G = (V, E)$ with edge weights, and the goal is to partition V into two groups to maximize the total weight of the edges with one endpoint in each group. Consider a local search algorithm that modifies the current solution by moving a single vertex from one side to the other (known as the "flip neighborhood"), and performs such moves as long as they increase the sum of the weights of the edges crossing the cut. In the worst case, this local search algorithm can require an exponential number of iterations to converge. What about in the smoothed analysis model, where a small random perturbation is added

[m] The extensively studied "stochastic block model" generalizes the planted clique model (for example, see Moore[32]), and is another fruitful playground for semi-random models. Here, the vertices of a graph are partitioned into groups, and the probability that an edge is present is a function of the groups that contain its endpoints. The responsibility of an algorithm in this model is to recover the (unknown) vertex partition. This goal becomes provably strictly harder in the presence of a monotone adversary.[31]

[n] This perturbation results in a dense constraint matrix even if the original one was sparse, and for this reason Theorem 3 is not fully satisfactory. Extending this result to sparsity-preserving perturbations is an important open question.

[o] An orthogonal issue with local search heuristics is the possibility of outputting a locally optimal solution that is much worse than a globally optimal one. Here, the gap between theory and practice is not as embarrassing—for many problems, local search algorithms really can produce pretty lousy solutions. For this reason, one generally invokes a local search algorithm many times with different starting points and returns the best of all of the locally optimal solutions found.

to each edge's weight? The natural conjecture is that local search should terminate in a polynomial number of iterations, with high probability over the perturbation. This conjecture has been proved for graphs with maximum degree $O(\log n)$[21] and for the complete graph;[4] for general graphs, the state-of-the-art is a quasi-polynomial-time guarantee (meaning $n^{O(\log n)}$ iterations).[22]

More ambitiously, it is tempting to speculate that for every natural local search problem, local search terminates in a polynomial number of iterations in the smoothed analysis model (with high probability). Such a result would be a huge success story for smoothed analysis and beyond worst-case analysis more generally.

On Machine Learning

Much of the present and future of research going beyond worst-case analysis is motivated by advances in machine learning.[p] The unreasonable effectiveness of modern machine learning algorithms has thrown down the gauntlet to algorithms researchers, and there is perhaps no other problem domain with a more urgent need for the beyond worst-case approach.

To illustrate some of the challenges, consider a canonical supervised learning problem, where a learning algorithm is given a dataset of object-label pairs and the goal is to produce a classifier that accurately predicts the label of as-yet-unseen objects (for example, whether or not an image contains a cat). Over the past decade, aided by massive datasets and computational power, deep neural networks have achieved impressive levels of performance across a range of prediction tasks.[25] Their empirical success flies in the face of conventional wisdom in multiple ways. First, most neural network training algorithms use first-order methods (that is, variants of gradient descent) to solve nonconvex optimization problems that had been written off as computationally intractable. Why do these algorithms so often converge quickly to a local optimum, or even to a global optimum?[q] Second, modern neural networks are typically over-parameterized, meaning that the number of free parameters (weights and biases) is considerably larger than the size of the training dataset. Over-parameterized models are vulnerable to large generalization error (that is, overfitting), but state-of- the-art neural networks generalize shockingly well.[40] How can we explain this? The answer likely hinges on special properties of both real-world datasets and the optimization algorithms used for neural network training (principally stochastic gradient descent).[r]

There are compelling parallels between the recent research on clustering in stable instances and slightly older results in a field of applied mathematics known as *sparse recovery,* where the goal is to reverse engineer a "sparse" object from a small number of clues around it.

Another interesting case study, this time in unsupervised learning, concerns topic modeling. The goal here is to process a large unlabeled corpus of documents and produce a list of meaningful topics and an assignment of each document to a mixture of topics. One computationally efficient approach to the problem is to use a singular value decomposition subroutine to factor the term-document matrix into two matrices, one that describes which words belong to which topics, and one indicating the topic mixture of each document.[35] This approach can lead to negative entries in the matrix factors, which hinders interpretability. Restricting the matrix factors to be nonnegative yields a problem that is *NP*-hard in the worst case, but Arora et al.[6] gave a practical factorization algorithm for topic modeling that runs in polynomial time under a reasonable assumption about the data. Their assumption states that each topic has at least one "anchor word," the presence of which strongly indicates that the document is at least partly about that topic (such as the word "Durant" for the topic "basketball"). Formally articulating this property of data was an essential step in the development of their algorithm.

The beyond worst-case viewpoint can also contribute to machine learning by "stress-testing" the existing theory

[p] Arguably, even the overarching goal of research in beyond worst-case analysis—determining the best algorithm for an application-specific special case of a problem—is fundamentally a machine learning problem.[26]

[q] See Jin et al.[28] and the references therein for recent progress on this question.

[r] See Neyshabur[33] and the references therein for recent developments in this direction.

and providing a road map for more robust guarantees. While work in beyond worst-case analysis makes strong assumptions relative to the norm in theoretical computer science, these assumptions are usually weaker than the norm in statistical machine learning. Research in the latter field often resembles average-case analysis, for example when data points are modeled as independent and identically distributed samples from some (possibly parametric) distribution. The semi-random models described earlier in this article are role models in blending adversarial and average-case modeling to encourage the design of algorithms with robustly good performance. Recent progress in computationally efficient robust statistics shares much of the same spirit.[19]

Conclusion

With algorithms, silver bullets are few and far between. No one design technique leads to good algorithms for all computational problems. Nor is any single analysis framework—worst-case analysis or otherwise—suitable for all occasions. A typical algorithms course teaches several paradigms for algorithm *design*, along with guidance about when to use each of them; the field of beyond worst-case analysis holds the promise of a comparably diverse toolbox for algorithm *analysis*.

Even at the level of a specific problem, there is generally no magical, always-optimal algorithm—the best algorithm for the job depends on the instances of the problem most relevant to the specific application. Research in beyond worst-case analysis acknowledges this fact while retaining the emphasis on robust guarantees that is central to worst-case analysis. The goal of work in this area is to develop novel methods for articulating the relevant instances of a problem, thereby enabling rigorous explanations of the empirical performance of known algorithms, and also guiding the design of new algorithms optimized for the instances that matter.

With algorithms increasingly dominating our world, the need to understand when and why they work has never been greater. The field of beyond worst-case analysis has already produced several striking results, but there remain many unexplained gaps between the theoretical and empirical performance of widely used algorithms. With so many opportunities for consequential research, I suspect the best work in the area is yet to come.

Acknowledgments. I thank Sanjeev Arora, Ankur Moitra, Aravindan Vijayaraghavan, and four anonymous reviewers for several helpful suggestions. This work was supported in part by NSF award CCF-1524062, a Google Faculty Research Award, and a Guggenheim Fellowship. This article was written while the author was at Stanford University.

References

1. Ackerman, M. and Ben-David, S. Clusterability: A theoretical study. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, 2009, 1–8.
2. Albers, S., Favrholdt, L.M. and Giel, O. On paging with locality of reference. *J. Computer and System Sciences 70*, 2 (2005), 145–175.
3. Alon, N., Krivelevich, M. and Sudakov, B. Finding a large hidden clique in a random graph. *Random Structures & Algorithms 13*, 3-4 (1998), 457–466.
4. Angel, O., Bubeck, S., Peres, Y., and Wai, F. Local MAX-CUT in smoothed polynomial time. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, 2017, 429-437.
5. Angelidakis, H., Makarychev, K., and Makarychev, Y. Algorithms for stable and perturbation-resilient problems. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, pages 438-451, 2017.
6. Arora, S., Ge, R., Halpern, Y. Mimno, D.M., Moitra, A., Sontag, D., Wu, Y. and Zhu, M. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the 30th International Conference on Machine Learning*, 2013, 280–288.
7. Arthur, D., Manthey, B., and Roglin, H. Smoothed analysis of the *k*-means method. *JACM 58*, 5 (2011), Article No. 18.
8. Awasthi, P., Blum, A. and Sheffet, O. Center-based clustering under perturbation stability. *Information Processing Letters 112*, 1–2 (2012), 49–54.
9. Balcan, M.-F., Blum, A. and Gupta, A. Clustering under approximation stability. *JACM 60*, 2 (2013), Article No. 8.
10. Balcan, M.-F., Haghtalab, N. and White, C. *k*-center clustering under perturbation resilience. In *Proceedings of the 43rd Annual International Colloquium on Automata, Languages, and Programming*, 2016, Article No. 68.
11. Barak, B., Hopkins, S.B., Kelner, J.A., Kothari, P., Moitra, A., and Potechin, A. A nearly tight sum-of squares lower bound for the planted clique problem. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, 2016, 428–437.
12. Bilu, Y. and Linial, N. Are stable instances easy? *Combinatorics, Probability & Computing 21*, 5 (2012), 643–660.
13. Blum, A. and Spencer, J.H. Coloring random and semi-random *k*-colorable graphs. *J. Algorithms 19*, 2 (1995), 204–234.
14. Borgwardt, K.H. *The Simplex Method: A probabilistic analysis.* Springer-Verlag, 1980.
15. Candes, E.J., Romberg, J.K., and Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete Fourier information. *IEEE Trans. Information Theory 52*, 2 (20006), 489–509.
16. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M, and Saurabh, S. *Parameterized Algorithms.* Springer, 2015.
17. Dadush, D. and Huiberts, S. A friendly smoothed analysis of the simplex method. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing*, 2018, 390–403.
18. Daniely, A., Linial, N. and M. Saks. Clustering is difficult only when it does not matter. arXiv:1205.4891, 2012.
19. Diakonikolas, I., Kamath, G., Kane, D.M., Li, J., Moitra, A, and Stewart, A. Being robust (in high dimensions) can be practical. In *Proceedings of the 34th International Conference on Machine Learning*, 2017, 999–1008.
20. Donoho, D.L. Compressed sensing. *IEEE Trans. Information Theory 52*, 4 (2006), 1289–1306.
21. Elsasser, R. and Tscheuschner, T. Settling the complexity of local max-cut (almost) completely. In *Proceedings of the 38th Annual International Colloquium on Automata, Languages, and Programming*, 2011, 171-182.
22. Etscheid, M. and Roglin, H. Smoothed analysis of local search for the maximum-cut problem. *ACM Trans. Algorithms 13*, 2 (2017), Article No. 12.
23. Fagin, R., Lotem, A. and Naor, M. Optimal aggregation algorithms for middleware. *J. Computer and System Sciences 66*, 4 (2003), 614–656.
24. Feige, U. and Kilian, J. Heuristics for semirandom graph problems. *J. Computer and System Sciences 63*, 4 (2001), 639–671.
25. Goodfellow, I., Bengio, Y. and Courville, A. *Deep Learning.* MIT Press, 2016.
26. Gupta, R. and Roughgarden, T. Application-specific algorithm selection. *SIAM J. Computing 46*, 3 (2017), 992–1017.
27. Jerrum, M. Large cliques elude the Metropolis process. *Random Structures and Algorithms 3*, 4 (1992), 347–359.
28. Jin, C, Ge, R., Netrapalli, P., Kakade, S.M. and Jordan, M.I. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning*, 2017, 1724–1732.
29. Kumar, A. and Kannan, R. Clustering with spectral norm and the *k*-means algorithm. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, 2010, 299–308.
30. Makarychev, K., Makarychev, Y. and Vijayaraghavan, A. Bilu-Linial stable instances of max cut and minimum multiway cut. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2014, 890-906.
31. Moitra, A., Perry, W. and Wein, A.S. How robust are reconstruction thresholds for community detection? In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, 2016, 828–841.
32. Moore, C. The computer science and physics of community detection: Landscapes, phase transitions, and hardness. *Bulletin of EATCS 121* (2017), 1–37.
33. Neyshabur, B. *Implicit Regularization in Deep Learning.* Ph.D. thesis, Toyota Technological Institute at Chicago, 2017.
34. Ostrovsky, R., Rabani, Y., Schulman, L.J. and Swamy, C. The effectiveness of Lloyd-type methods for the *k*-means problem. *JACM 59*, 6 (2012), Article No. 22.
35. Papadimitriou, C.H., Raghgavan, P., Tamaki, H. and Vemapala, S. Latent semantic indexing: A probabilistic analysis. *J. Computer and System Sciences 61*,2 (2000), 217–235.
36. Roughgarden, T. CS264 lecture notes on beyond worst-case analysis. Stanford University, 2009–2017. Available at http://www.timroughgarden.org/notes.html.
37. Sleator, D.D. and Tarjan, R.E. Amortized efficiency of list update and paging rules. *Commun. ACM 28*, 2 (1985), 202–208.
38. Spielman, D.A. and Teng, S.-H. Smoothed analysis: Why the simplex algorithm usually takes polynomial time. *JACM 51*, 3 (2004), 385–463.
39. Trevisan, L. Inapproximability of combinatorial optimization problems. *Paradigms of Combinatorial Optimization: Problems and New Approaches.* V.T. Paschos, ed. Wiley, 2014.
40. Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O. Understanding deep learning requires rethinking generalization. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Tim Roughgarden is a professor in the computer science department at Columbia University, New York, NY, USA.

Watch the author discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/beyond-worst-case-analysis

research highlights

DOI:10.1145/3306206

To view the accompanying paper, visit doi.acm.org/10.1145/3306204 rh

Technical Perspective
Borrowing Big Code to Automate Programming Activities

By Martin C. Rinard

BIG DATA COMBINED with machine learning has revolutionized fields such as computer vision, robotics, and natural language processing. In these fields, automated techniques that detect and exploit complex patterns hidden within large datasets have repeatedly outperformed techniques based on human insight and intuition.

But despite the availability of enormous amounts of code (big code) that could, in theory, be leveraged to deliver similar advances for software, programming has proved to be remarkably resistant to this kind of automation. Much programming today consists of developers deploying keyword searches against online information aggregators such as Stack Overflow to find, then manually adapt, code sequences that implement desired behaviors.

The following paper presents new techniques for leveraging big code to automate two programming activities: selecting understandable names for JavaScript identifiers and generating type annotations for JavaScript variables. The basic approach leverages large JavaScript code bases to build a probabilistic model that predicts names and type annotations given the surrounding context (which includes constants, JavaScript API calls, and variable uses in JavaScript expressions and statements).

When run on programs with the original variable names obfuscated, the implemented system was able to recover the original variable names over 60% of the time. The results for type annotations are even more intriguing—the implemented system generates correct type annotations for over half of the benchmark programs. For comparison, the programmer-provided annotations are correct for only a bit over a quarter of these programs. The system is accessible via the Internet at jsnice.org with hundreds of thousands of users.

These results demonstrate how this approach can help JavaScript programmers produce more easily readable and understandable programs. One potential longer-range consequence could be the gradual emergence of a de facto standard for aspects of JavaScript programs such as variable names and the relationship between program structure and types. More broadly, the results also highlight the substantial redundancy present in JavaScript code worldwide and raise questions about just how much human effort is really required to produce this code.

Why was this research so successful? First, the authors chose a problem that was a good fit for machine learning over big code. Current machine learning techniques do not provide correct results; they instead only provide results that look like previous results in the training set. A variable name or type annotation predictor does not have to always be correct; it only needs to be correct enough of the time to be useful. And JavaScript programs share enough variable name and type annotation patterns to support a reasonably accurate model.

A second reason is technical, specifically the development of a program representation that exposes relevant relationships between variables and the surrounding context, including how variables are used in JavaScript statements and expressions. Features exposed in this program representation enable the immediate application of conditional random fields, a standard technique in machine learning for structured prediction previously shown to be effective for solving problems in areas such as natural language processing and computer vision, to solve the learning and prediction problem. The development of a new approximate MAP inference algorithm for this domain enables the performance required for interactive use when working with thousands of labels per node (in contrast to many previous applications, which only work with tens of labels per node).

Why was this research so successful? First, the authors chose a problem that was a good fit for machine learning over big code.

What can we expect to see in the future from this line of research? The most obvious next steps include a variety of automated programming assistants for tasks such as code search, code completion, and automatic patch generation. Here the assistant would interact with the programmer to guide the process of turning vague, uncertain, or underspecified goals into partially or fully realized code, with programmer supervision required to complete and/or ensure the correctness of the resulting code.

It is less clear how to make progress on programming tasks with more demanding correctness, autonomy, or novelty requirements. One critical step may be finding productive ways to integrate probabilistic reasoning with more traditional logical reasoning as applied to computer programs. Future research, potentially inspired in part by the results presented in this paper, will determine the feasibility of this goal. C

Martin C. Rinard is a professor in the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology, Cambridge, MA, USA, and a member of the Computer Science and Artificial Intelligence Laboratory.

DOI:10.1145/3306204

Predicting Program Properties from 'Big Code'

By Veselin Raychev, Martin Vechev, and Andreas Krause

Abstract

We present a new approach for predicting program properties from large codebases (aka "Big Code"). Our approach learns a probabilistic model from "Big Code" and uses this model to predict properties of new, unseen programs.

The key idea of our work is to transform the program into a representation that allows us to formulate the problem of inferring program properties as structured prediction in machine learning. This enables us to leverage powerful probabilistic models such as Conditional Random Fields (CRFs) and perform *joint* prediction of program properties.

As an example of our approach, we built a scalable prediction engine called JSNice for solving two kinds of tasks in the context of JavaScript: predicting (syntactic) names of identifiers and predicting (semantic) type annotations of variables. Experimentally, JSNice predicts correct names for 63% of name identifiers and its type annotation predictions are correct in 81% of cases. Since its public release at http://jsnice.org, JSNice has become a popular system with hundreds of thousands of uses.

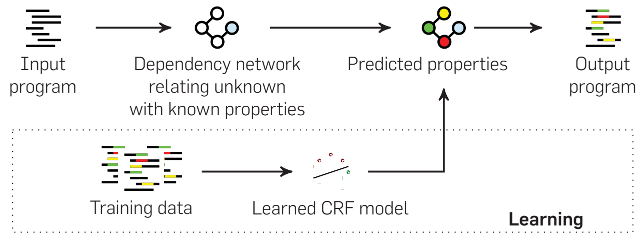
By formulating the problem of inferring program properties as structured prediction, our work opens up the possibility for a range of new "Big Code" applications such as de-obfuscators, decompilers, invariant generators, and others.

1. INTRODUCTION

Recent years have seen significant progress in the area of programming languages driven by advances in type systems, constraint solving, program analysis, and synthesis techniques. Fundamentally, these methods reason about each program in *isolation* and while powerful, the effectiveness of programming tools based on these techniques is approaching its inherent limits. Thus, a more disruptive change is needed if a significant improvement is to take place.

At the same time, creating probabilistic models from large datasets (also called "Big Data") has transformed a number of areas such as natural language processing, computer vision, recommendation systems, and many others. However, despite the overwhelming success of "Big Data" in a variety of application domains, learning from large datasets of programs has previously not had tangible impact on programming tools. Yet, with the tremendous growth of publicly available source code in repositories such as GitHub[4] and BitBucket[2] (referred to as "Big Code" by a recent DARPA initiative[11]) comes the opportunity to create new kinds of programming tools based on probabilistic models of such data. The vision is that by leveraging the massive effort already spent in developing millions of programs, such tools will have the ability to solve tasks beyond the reach of traditional techniques. However, effectively learning from programs is a challenge. One reason is that programs are data *transformers* with complex semantics that should be captured and preserved in the learned probabilistic model.

Figure 1. Structured prediction for programs.



1.1. Structured prediction for programs

The core technical insight of our work is transforming the input program into a representation that enables us to formulate the problem of predicting program properties as structured prediction with Conditional Random Fields (CRFs).[15] Indeed, CRFs are a powerful probabilistic model successfully used in a wide variety of applications including computer vision and natural language processing.[12, 15, 16] We show how to instantiate this approach towards predicting semantic information (e.g., type annotations) as well as syntactic facts (e.g., identifier names). To our knowledge, this is the first work which shows how CRFs can be learned and used in the context of programs. By connecting programs to CRFs, a wide range of learning and inference algorithms[14] can be used in the domain of programs.

Figure 1 illustrates the structured prediction approach. In the prediction phase (upper part of figure), we are given an input program for which we are to infer properties of interest. In the next step, we convert the program into a representation called dependency network. The dependency network captures relationships between program elements whose properties are to be predicted with elements whose properties are known. Once the network is obtained, we perform structured prediction and in particular, a query referred to as Maximum a Posteriori (MAP) inference.[14] This query makes a *joint* prediction for all elements together by optimizing a scoring function based on the learned CRF model. Making a joint prediction which takes into account structure and dependence is particularly important as

The original version of this paper was published in ACM POPL'2015.

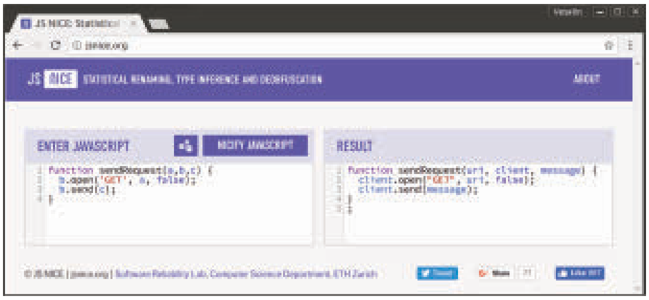
properties of different elements are often related. A useful analogy is the ability to make joint predictions in image processing where the prediction of a pixel label is influenced by the predictions of neighboring pixels.

1.2. JSNice: Name and type inference for JavaScript

As an example of this approach, we built a system which addresses two important challenges in JavaScript: predicting (syntactic) identifier names and (semantic) type annotations of variables. Such predictions have applications in software engineering (e.g., refactoring to improve code readability), program analysis (e.g., type inference) and security (e.g., deobfuscation). We focused on JavaScript for three reasons. First, in terms of type inference, recent years have seen extensions of JavaScript that add type annotations such as the Google Closure Compiler[5] and TypeScript.[7] However, these extensions rely on traditional type inference, which does not scale to realistic programs that make use of dynamic evaluation and complex libraries (e.g., jQuery).[13] Our work predicts likely type annotations for real world programs which can then be provided to the programmer or to a standard type checker. Second, much of JavaScript code found on the Web is obfuscated, making it difficult to understand what the program is doing. Our approach recovers likely identifier names, thereby making much of the code on the Web readable again. This is enabled by a large and well-annotated corpus of JavaScript programs available in open source repositories such as GitHub.

Since its release, JSNice has become a widely used system with users ranging from JavaScript developers to security specialists. In a period of a year, our users deobfuscated over 9 GB (87.7 mn lines of code) of unique (non-duplicate) JavaScript programs. Figure 3 shows a histogram of the size of these programs, indicating that users often query it with large code fragments. The average JavaScript program size is 91.7 KB.

Figure 2. A screenshot of http://jsnice.org/: minified code (left), deobfuscated version (right).



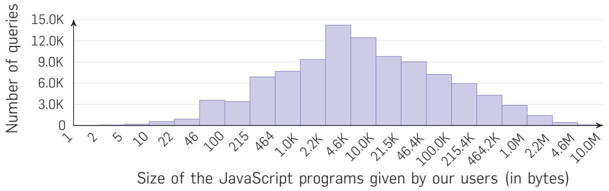
JS NICE: Statistical
jsnice.org
JS NICE STATISTICAL RENAMING, TYPE INFERENCE AND DEOBFUSCATION
ABOUT
ENTER JAVASCRIPT
NICIFY JAVASCRIPT
RESULT

```
function sendRequest(a,b,c) {
  b.open('GET', a, false);
  b.send(c);
}
```

```
function sendRequest(url, client, message) {
  client.open("GET", url, false);
  client.send(message);
}
;
```

© JS NICE | jsnice.org | Software Reliability Lab, Computer Science Department, ETH Zurich

Figure 3. Histogram of query sizes to http://jsnice.org/ sent by users in the period May 10, 2015–May 10, 2016.



1.3. Nice2Predict: Structured prediction framework

To facilitate faster creation of new applications (JSNice being one example), we built a reusable framework called Nice2Predict (found at http://nice2predict.org) which includes all components of this work (e.g., training and inference) except the definition of feature functions (which are application specific). Then, to use our method one only needs to phrase their application in terms of a CRF model which is done by defining suitable feature functions (we show such functions for JSNice later in the paper) and then invoke the Nice2Predict training and inference mechanisms. A recent example of this instantiation is DeGuard[9] (http://apk-deguard.com), a system that performs Android layout de-obfuscation by predicting method, class, and field names erased by ProGuard.[6]

2. OVERVIEW

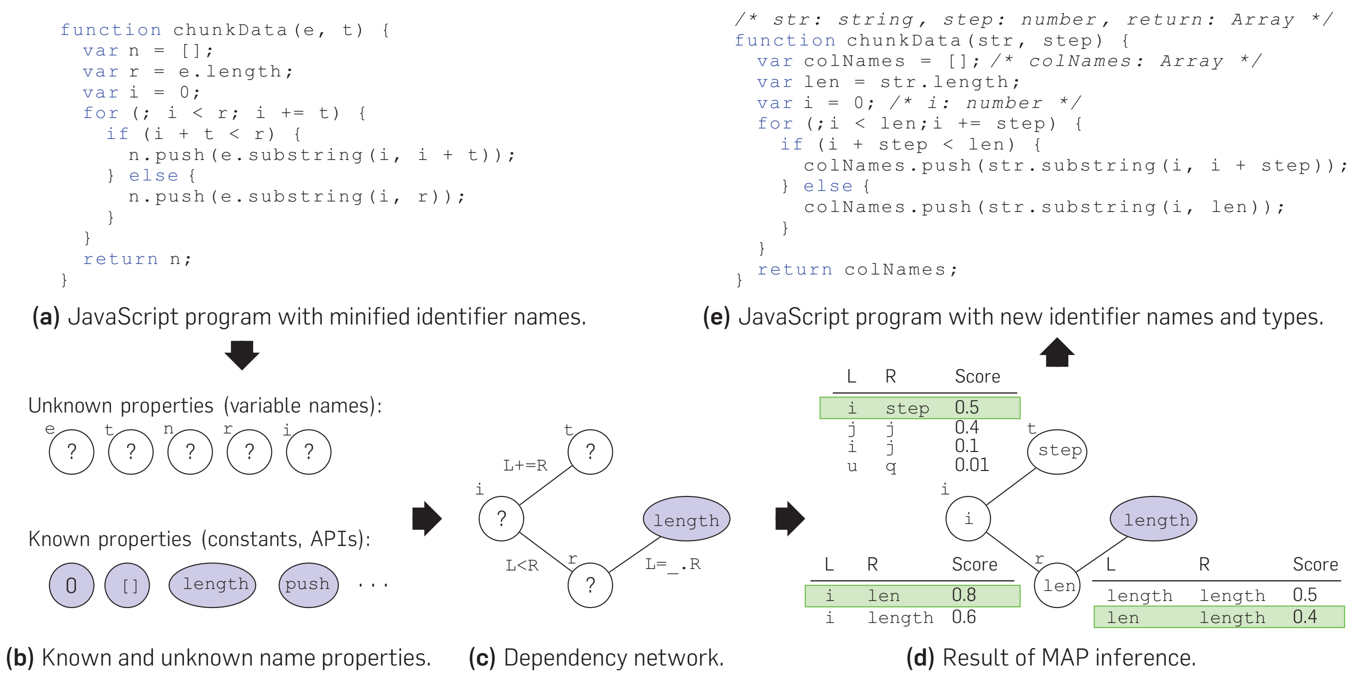
We now provide an informal description of our probabilistic approach on a running example. Consider the JavaScript program shown in Figure 4(a). This is a program which has short, non-descriptive identifier names. Such names can be produced by both a novice inexperienced programmer or by an automated process known as minification (a form of layout obfuscation) which replaces identifier names with shorter names. In the case of client-side JavaScript, minification is a common process on the Web and is used to reduce the size of the code being transferred over the network and/or to prevent users from understanding what the program is actually doing. In addition to obscure names, variables in this program also lack annotated type information. It can be difficult to understand that this obfuscated program happens to partition an input string into chunks of given sizes, storing those chunks into consecutive entries of an array.

Given the program in Figure 4(a), JSNice automatically produces the program in Figure 4(e). The output program has new identifier names and is annotated with predicted types for the parameters, local variables, and return statement. Overall, it is easier to understand what that program does when compared to the original. We now provide an overview of the prediction recipe that performs this transformation. We focus on predicting names (reversing minification), but the process for predicting types is identical.

2.1. Step 1: Determine known and unknown properties

Given the program in Figure 4(a), we use a simple static (scope) analysis which determines the set of program elements for which we would like to infer properties. These are elements whose properties are *unknown* in the input (i.e., are affected by minification). When predicting names, this set consists of all local variables and function parameters of the input program: `e`, `t`, `n`, `r`, and `i`. We also determine the set of elements whose properties are *known* (not affected by minification). These include field and method names (e.g., the field element with name `length`). Both kinds of elements are shown in Figure 4(b). The goal is to predict the *unknown* properties based on: (i) the obtained

Figure 4. Probabilistic inference of program properties on an example.



```
function chunkData(e, t) {
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t) {
    if (i + t < r) {
      n.push(e.substring(i, i + t));
    } else {
      n.push(e.substring(i, r));
    }
  }
  return n;
}
```

(a) JavaScript program with minified identifier names.

```
/* str: string, step: number, return: Array */
function chunkData(str, step) {
  var colNames = []; /* colNames: Array */
  var len = str.length;
  var i = 0; /* i: number */
  for (;i < len;i += step) {
    if (i + step < len) {
      colNames.push(str.substring(i, i + step));
    } else {
      colNames.push(str.substring(i, len));
    }
  }
  return colNames;
}
```

(e) JavaScript program with new identifier names and types.

Unknown properties (variable names):
e
t
n
r
i
?
?
?
?
?
Known properties (constants, APIs):
0
[]
length
push
...

(b) Known and unknown name properties.

(c) Dependency network.

L	R	Score
i	step	0.5
j	j	0.4
i	j	0.1
u	q	0.01

L	R	Score
i	len	0.8
i	length	0.6

L	R	Score
length	length	0.5
len	length	0.4

(d) Result of MAP inference.

known properties and (ii) the relationship between elements (discussed here).

2.2. Step 2: Build dependency network

Next, using features we later define, we build a dependency network capturing relationships between program elements. The dependency network is key to capturing structure when performing predictions and intuitively determines how properties to be predicted influence each other. For example, the link between known and unknown properties allows us to leverage the fact that many programs use common anchors (e.g., JQuery API) meaning that the unknown quantities we aim to predict are influenced by how known elements are used. Dependencies are triplets of the form ⟨*n*,*m*,*rel*⟩ where, *n, m* are program elements, and *rel* is the particular relationship between the two elements. In our work all dependencies are triplets, but these can be extended to relationships involving more than two elements.

In Figure 4(c), we show three example dependencies between the program elements. For instance, the statement `i += t` generates a dependency ⟨`i`,`t`,`L+=R`⟩, because `i` and `t` are on the left and right side of a`+=` expression. Similarly, the statement `var r = e.length` generates several dependencies including ⟨`r`,`length`,`L=_.R`⟩ which designates that the left part of the relationship, denoted by `L`, appears before the de-reference of the right side denoted by `R` (we elaborate on the different types of relationships in Section 4). For clarity, in Figure 4(c) we include only some of the relationships.

2.3. Step 3: MAP inference

After obtaining the dependency network, we next infer the most likely values for the nodes via a query known as MAP inference.[14] Informally, in this type of query, a prediction leverages the structure of the network (i.e., the connections between the nodes) as opposed to predicting separately each node at a time. As illustrated in Figure 4(d), for the network of Figure 4(c), our system infers the new names `step` and `len`. It also predicts that the previous name `i` was most likely. Let us consider how we predicted the names `step` and `len`. The network in Figure 4(d) is the same one as in Figure 4(c) but with additional tables produced as an output of the learning phase (i.e., the learning determines the concrete feature functions and the weights associated with them). Each table is a function that scores the assignment of properties when connected by the corresponding edge (intuitively, how likely the particular pair is). Consider the topmost table in Figure 4(d). The first row says the assignment of `i` and `step` is scored with 0.5. The MAP inference searches for assignment of properties to nodes such that the *sum* of the scores shown in the tables is maximized. For the two nodes `i` and `t`, the inference ends up selecting the highest score from that table (i.e., the values `i` and `step`). Similarly for nodes `i` and `r`. However, for nodes `r` and `length`, the inference *does not* select the topmost row but the values from the second row. The reason is that if it had selected the topmost row, then the only viable choice (in order to match the value `length`) for the remaining relationship is the second row of `i, r` table (with value 0.6). However, the assignment 0.6 leads to a lower *combined* overall score. That is, MAP inference must take into account the global *structure* and cannot simply select the maximal score of each function.

To achieve good performance for the MAP inference, we developed a new algorithmic variant which targets the domain of programs (existing inference algorithms cannot efficiently deal with the combination of unrestricted network structure and a large number of possible predictions per element).

2.4. Output program

Finally, after the new names are inferred, our system transforms the original program to use these names. The output of the entire inference process is captured in the program shown in Figure 4(e). Notice how in this output program, the names tend to accurately capture what the program does.

2.5. Predicting type annotations

Even though we illustrated the inference process for variable names, the overall flow for predicting type annotations is identical. Interestingly, after predicting types, one can invoke a standard type checker to check whether the predicted types are valid for the program. For our example in Figure 4(e), the predicted type annotations (shown in comments) are indeed valid. In general, when predicting semantic properties (such as types) where soundness plays a role, our approach can be used as part of a guess-and-check loop.

2.6. Independent of minification

We note that our name inference process is independent of what the minified names are. In particular, the process will return the same names regardless of which minifier was used to obfuscate the original program (provided these minifiers always rename the *same set* of variables).

3. STRUCTURED PREDICTION

We now introduce the structured prediction approach. We later instantiate this approach in Section 4.

3.1. Notation: programs, labels, predictions

Let $x \in X$ be a program. As with standard program analysis, we will infer properties about program statements or expressions (referred to as program elements). For a program x, each element (e.g., a variable) is identified with an index (a natural number). We usually separate the elements into two kinds: (i) elements for which we are interested in inferring properties and (ii) elements whose properties we know (e.g., obtained via say standard program analysis or manual annotation). We use two helper functions n, m: $X \rightarrow \mathbb{N}$ to return the appropriate number of program elements for a given program x: $n(x)$ returns the number of elements of kind (i) and $m(x)$ the number of elements of kind (ii). When x is clear from the context, we write n instead of $n(x)$ and m instead of $m(x)$.

We use the set $Labels_U$ to denote all possible values that a predicted property can take. For instance, in type prediction, $Labels_U$ contains all possible basic types (e.g., number, string, etc). Then, for a program x, we use the notation $\boldsymbol{y} = (y_1, \ldots, y_{n(x)})$ to denote a vector of predicted program properties. Here, $\boldsymbol{y} \in Y$ where $Y = (Labels_U)^*$. That is, each entry y_i in vector $\boldsymbol{y}$ ranges over $Labels_U$ and denotes that program element i has property y_i.

For a program x, we define the vector $\boldsymbol{z}^x = (z_1^x, \ldots, z_\mathrm{m}^x)$ to capture known properties. Here, each z_i^x can take a value from the set of properties $Labels_K$ which could potentially differ from the set of properties $Labels_U$ that we use for prediction. For example, if the known properties are integer constants, $Labels_K$ will contain all valid integers. To avoid clutter when x is clear from the context, we use $\boldsymbol{z}$ instead of $\boldsymbol{z}^x$. We use $Labels = Labels_U \cup Labels_K$ to denote the set of all properties.

Note that to apply this method the total number of predictions must be fixed (bounded) in advance (i.e., $n(x)$). This is unlike other settings, for example, grammars,[10] where the number of predictions can be unbounded.

3.2. Problem definition

Let $D = \{\langle x^{(j)}, \boldsymbol{y}^{(j)} \rangle\}_{j=1}^{t}$ denote the training data: a set of t programs each annotated with corresponding properties. Our goal is to learn a model that captures the conditional probability $Pr(\boldsymbol{y} \mid x)$. Once the model is learned, we can predict properties of new programs by posing the following MAP query:

Given a **new** program x, find $\boldsymbol{y} = \arg\max_{\boldsymbol{y}' \in \Omega_x} Pr(\boldsymbol{y}' \mid x)$

That is, we aim to find the most likely assignment of program properties $\boldsymbol{y}$ according to the probabilistic model. Here, $\Omega_x \subseteq Y$ describes the set of possible assignments of properties $\boldsymbol{y}'$ to program elements of x. The set Ω_x allows restricting the set of possible properties and is useful for encoding problem-specific constraints. For example, in type annotation inference, the set Ω_x may restrict the annotations to types that make the resulting program typecheck.

3.3. Conditional random fields (CRFs)

We now briefly describe CRFs, a particular model defined in Lafferty et al.[15] and previously used for a range of tasks such as natural language processing, image classification, and others. CRFs represent the conditional probability $Pr(\boldsymbol{y} \mid x)$. We consider the case where the factors are positive in which case, without loss of generality, any conditional probability of properties $\boldsymbol{y}$ given a program x can be encoded as follows:

$$Pr(\boldsymbol{y} \mid x) = \frac{1}{Z(x)} \exp(score(\boldsymbol{y}, x))$$

where, *score* is a function that returns a real number indicating the score of an assignment of properties $\boldsymbol{y}$ for a program x. Assignments with higher score are more likely than assignments with lower score. $Z(x)$, called the *partition function*, ensures that the above expression does in fact encode a conditional distribution. It returns a real number depending only on the program x, that is:

$$Z(x) = \sum_{\boldsymbol{y}' \in \Omega_x} \exp(score(\boldsymbol{y}', x))$$

W.l.o.g, *score* can be expressed as a composition of a sum of k feature functions f_i associated with weights w_i:

$$score(\boldsymbol{y}, x) = \sum_{i=1}^{k} w_i f_i(\boldsymbol{y}, x) = \boldsymbol{w}^T \boldsymbol{f}(\boldsymbol{y}, x)$$

Here, $\boldsymbol{f}$ is a vector of functions f_i and $\boldsymbol{w}$ is a vector of weights

w_i. The feature functions f_i: $Y \times X \rightarrow \mathbb{R}$; are used to score assignments of program properties. This representation of score functions is well-suited for learning (as the weights $\boldsymbol{w}$ can be learned from data).

3.4. Features as constraints

Feature functions are key to controlling the likelihood predictions. For instance, a feature function can be defined to prohibit or lower the score of an undesirable prediction: say if $f_i(\boldsymbol{y}^B, x) = -H$ where H is a very large positive number, then f_i (with weight $w_i > 0$) essentially disables an assignment $\boldsymbol{y}^B$ as $Pr(\boldsymbol{y}^B \mid x)$ will approach 0.

3.5. General prediction approach

Let us first define the kind of relations between program elements we use when making predictions. Let the set of possible relations be *Rels*. An example relation we considered in our running example was `L+=R`. It relates variables `i` and `t` in Figure 4(c) and arises due to the expression `i+=t` in the code. Examples of other relations are found in Section 4.

3.6. Pairwise indicator feature functions

Let $\{\psi_i\}_{i=1}^{k}$ be a set of pairwise feature functions where each ψ_i: $Labels \times Labels \times Rels \rightarrow \mathbb{R}$ scores a *pair* of properties when related with a relation from *Rels*. For example:

$$\psi_{ex}(l_1, l_2, e) = \begin{cases} 1 & \textbf{if } l_1 = \texttt{i} \textbf{ and } l_2 = \texttt{step} \textbf{ and } e = \texttt{L+=R} \\ 0 & \textbf{otherwise} \end{cases}$$

In general, any feature function can be used, but our work shows that these pairwise functions are sufficient for making high-quality predictions of names and types. Next, we go over the steps of the prediction procedure more formally.

3.7. Step 1: Build dependency network

We begin by building the network $G^x = \langle V^x, E^x \rangle$ for a program x, capturing dependencies between predictions. Here, $V^x = V_U^x \cup V_K^x$ consists of elements (e.g., variables) for which we would like to predict properties V_U^x and elements whose properties we already know V_K^x. The set of edges $E^x \subseteq V^x \times V^x \times Rels$ denotes the fact that there is a relationship between two program elements and describes what that relationships is. This definition of network is also called a *multi-graph* because there is no restriction on having only a single edge between a pair of nodes – our definition permits multiple dependencies with different *Rels*.

We define the feature functions $\boldsymbol{f}(\boldsymbol{y}, x)$ over the graph G^x as follows. Let $(\boldsymbol{y}, \boldsymbol{z}^x)$ be a concatenation of the unknown properties $\boldsymbol{y}$ and the known properties $\boldsymbol{z}^x$ in x. Then, f_i is defined as the sum of the applications of its corresponding ψ_i over the set of all network edges in G^x:

$$f_i(\boldsymbol{y}, x) = \sum_{\langle a,b,rel\rangle \in E^x} \psi_i((\boldsymbol{y}, \boldsymbol{z}^x)_a, (\boldsymbol{y}, \boldsymbol{z}^x)_b, rel)$$

3.8. Step 2: MAP inference

Recall that the key query we perform is MAP inference. That is, given a program x, find a prediction $\boldsymbol{y}$ such that:

$$\boldsymbol{y} = \underset{\boldsymbol{y}' \in \Omega_x}{\arg\max}\ Pr(\boldsymbol{y}' \mid x)$$

As arg max is independent of $Z(x)$, we obtain an equivalent simplified query:

$$\boldsymbol{y} = \underset{\boldsymbol{y}' \in \Omega_x}{\arg\max}\ score(\boldsymbol{y}', x)$$

In theory, one can use any of the available inference algorithms to solve for the above query (exact inference is in general an NP-hard MaxSAT problem). In this work, we designed a fast and approximate greedy MAP inference algorithm tailored to our setting of programs: pairwise feature functions, unrestricted nature of G^x and the a large set of possible assignments. Our algorithm changes the labels of each node one-by-one or in pairs until the assignment score cannot be improved further.

3.9. Learning

The goal of the training phase (lower part of Figure 1) is to learn the weights $\boldsymbol{w}$ used in the *score* function from a large training set of programs. These weights cannot be obtained by means of counting in the training data.[14] [Section 20.3.1]. Instead, we use a learning technique from online support vector machines: given a training dataset $D = \{\langle x^{(j)}, \boldsymbol{y}^{(j)} \rangle\}_{j=1}^{t}$ of t samples, the goal is to find $\boldsymbol{w}$ such that the given assignments $\boldsymbol{y}^{(j)}$ are the highest scoring assignments in as many training samples as possible subject to additional margin learning constraints. The learning procedure is described in Ratliff et al.[17]

4. PREDICTING NAMES AND TYPES

In this section we present the JSNice system for prediting: (i) names of local variables and (ii) type annotations of function arguments. We investigate the above challenges in the context of JavaScript, a popular language where addressing the above two questions is of significant importance. We do note however that much of the machinery discussed in this section applies as-is to other programming languages.

4.1. JavaScript identifier name prediction

The goal of our name prediction task is to predict the (most likely) names of local variables in a given program x. We proceed as follows. First, we define V_K^x to range over all constants, objects properties, methods, and global variables of x. Each element in V_K^x can be assigned values from the set $Labels_K = JSConsts \cup JSNames$, where *JSNames* is a set of all valid identifier names and *JSConsts* is a set of possible constants. We note that object property names and Application Programming Interface (API) names are modeled as constants, as the dot (.) operator takes an object on the left-hand side and a string constant on the right-hand side. We define the set V_U^x to contain all local variables of x. Here, a variable name belonging to two different scopes leads to two program elements in V_U^x. Finally, $Labels_U$ ranges over *JSNames*.

To ensure the newly predicted names preserve program semantics, we ensure the following additional constraints hold: (i) *all* references to a renamed local variable must be renamed to the same name. This is enforced by how we define V_U^x (each element corresponds to a local variable as opposed to one element per variable occurrence), (ii) the predicted identifier names must not be reserved keywords. This is enforced by ensuring that $Labels_U$ does not contain keywords, and (iii) the prediction must not suggest the same name for two different variables in the same scope. This is enforced by prohibiting assignments of labels with conflicting names.

4.2. JavaScript type annotation prediction

Our second application involves probabilistic type annotation inference of function parameters. These annotations are particularly challenging to derive via standard program analysis techniques because such a derivation would require finding all possible callers of a function. Instead, we leverage existing manually (type) annotated JavaScript programs. In JSNICE we use JSDoc[1] annotated code for training data.

The simplified language over which we predict type annotations is defined as follows:

$$ex ::= val \mid var \mid ex_1(ex_2) \mid ex_1 \circledast ex_2 \qquad Expression$$
$$val ::= \lambda var : \tau.ex \mid n \qquad Value$$

Here, n ranges over constants ($n \in JSConsts$), var is a meta-variable ranging over the program variables, $*$ ranges over binary operators (+, -, *, /, ., <, ==, ===, etc.), and τ ranges over all possible variable types. That is, $\tau = \{?\} \cup L$ where L is a set of types (we discuss how to instantiate L below) and ? denotes the unknown type. To be explicit, we use the set $JSTypes$ where $JSTypes = \tau$. We use the function $[]_x : ex \rightarrow JSTypes$ to obtain the type of an expression in a program x. This map can be manually provided or built via program analysis. When the program x is clear from the context we use $[e]$ as a shortcut for $[]_x(e)$.

4.3. Defining known and unknown program elements

We define the set of unknown program elements as follows:

$$V_U^x = \{e \mid e \textbf{ is } var, [e] = ?\} \qquad Labels_U = JSTypes$$

That is, V_U^x contains variables whose type is unknown. In principle, we can differentiate between the type $\top$ and the unknown type ? in order to allow for finer control over which types we would like to predict. However, since standard type inference cannot predict types of function parameters, we annotate all non-annotated parameters with type ?.

Next, we define the set of known elements V_K^x. Note that V_K^x can contain any expression, not just variables like V_U^x:

$$V_K^x = \{e \mid e \textbf{ is } expr, [e] \neq ?\} \cup \{n \mid n \textbf{ is } constant\}$$
$$Labels_K = JSTypes \cup JSConsts$$

That is, V_K^x contains both, expressions whose types are known as well as constants. We do not restrict the set of possible assignments Ω_x, that is, $\Omega_x = (JSTypes)^n$ (recall that n is a function which returns the number of elements whose property is to be predicted). This means that we rely entirely on the learning to discover the rules that will produce non-contradicting types. The only restriction (discussed here) that we apply is constraining $JSTypes$ when performing predictions.

For $JSTypes = \{?\} \cup L$ the set L of types can be instantiated in various ways. In this work we define $L = \mathcal{P}(T)$ where $\langle T, \sqsubseteq \rangle$ is a complete lattice of types with T and $\sqsubseteq$ as defined in Figure 5. In the figure we use "..." to denote a potentially infinite number of user-defined object types. Note that L allows that a variable can be of a union type {string, number} which for convenience can also be written as string ∨ number.

4.4. Relating program elements

The relationships between program elements that we introduce define how to build the set of edges E^x of a program x. Since the elements for both prediction tasks are similar, so are the relationships. If a relationship is specific to a particular task, we explicitly state so in its description.

Relating expressions. The first relationship we discuss is syntactic in nature: it relates two program elements based on the program's Abstract Syntax Tree (AST). Let us consider the expression i+j<k. First, we build the AST of the expression as shown in Figure 6 (a). Suppose we are interested in performing name prediction for variables i, j, and k, represented with indices 1, 2, and 3 respectively, that is, $V_U^x = \{1,2,3\}$. Then, we build the dependency network as shown in Figure 6(b) to indicate the prediction for the three elements are dependent on one another (with the particular relationship shown over the edge). For example, the edge between 1 and 2 represents the relationship that these nodes participate in an expression L+R where L is a node for 1 and R is a node for 2.

Relationships are defined using the following grammar:

$$rel_{ast} ::= rel_L(rel_R) \mid rel_L \circledast rel_R$$
$$rel_L ::= \mathtt{L} \mid rel_L(\_) \mid \_(rel_L) \mid rel_L \circledast \_ \mid \_ \circledast rel_L$$
$$rel_R ::= \mathtt{R} \mid rel_R(\_) \mid \_(rel_R) \mid rel_R \circledast \_ \mid \_ \circledast rel_R$$

Figure 5. The lattice of types over which prediction occurs.

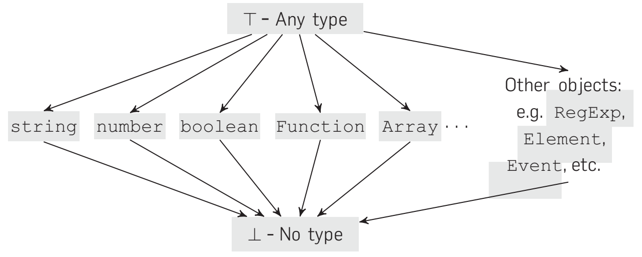
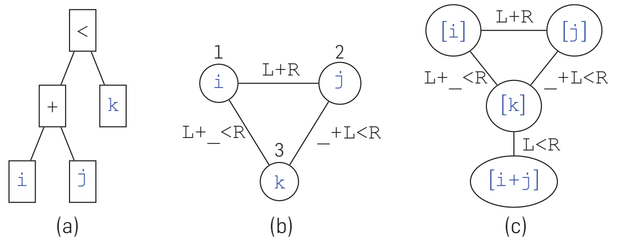


Figure 6. (a) The AST of expression i + j < k and two dependency networks built from the AST relations: (b) for name, and (c) for type predictions.



All relationships rel_{ast} are part of $Rels$, that is, $rel_{ast} \in Rels$. As discussed earlier, $\circledast$ ranges over binary operators. All relationships derived using the above grammar have exactly one occurrence of `L` and `R`. For a relationship $r \in rel_{ast}$, let $r[x/\mathtt{L}, y/\mathtt{R}, e/\_]$ denote the expression where x is substituted for `L`, y is substituted for `R` and the expression e is substituted for _. Then, given two program elements a and b and a relationship $r \in rel_{ast}$, a match is said to exist if $r[a/\mathtt{L}, b/\mathtt{R}, [expr]/\_] \cap Exp(x) \neq /$. Here, $[expr]$ denotes all possible expressions in the programming language and $Exp(x)$ is all expressions of program x. An edge $(a, b, r) \in E^x$ between two program elements a and b exists if there exists a match between a, b, and r.

Note that for a given pair of elements a and b there could be more than one relationship which matches, that is, both $r_1, r_2 \in rel_{ast}$ match where $r_1 \neq r_2$ (therefore, there could be multiple edges between a and b with different relationships).

The relationships described above are useful for both name and type inference. For names, the expressions being related are variables, while for types, they need not be restricted to variables. For example, in Figure 6(c) there is a relationship between the types of `k` and `i+j` via `L<R`. Note that our rules do not directly capture relationships between `[i]` and `[i+j]`, but they are transitively dependent. Still, many important relationships for type inference are present. For instance, in classic type inference, the relationship `L=R` implies a constraint rule $[\mathtt{L}] \sqsupseteq [\mathtt{R}]$ where $\sqsupseteq$ is the supertype relationship (indicated in Figure 5). Interestingly, our inference model can *learn* such rules instead of providing them manually.

Other relations. We introduce three other types of semantic relations: (i) a relationship capturing aliasing between expressions, (ii) a function calling relationship capturing whether a function (represented by a variable) may call another function (also represented by a variable), and (iii) an object access relationship specifying whether an object field (represented by a string constant) may be accessed by a function. The last two relationships are only used in name prediction and are particularly effective when predicting function names.

5. IMPLEMENTATION AND EVALUATION

We implemented our approach in an interactive system, called JSNice, which targets name and type predictions for JavaScript. JSNice modifies the Google Closure Compiler.[5] In standard operation, this compiler takes human-readable JavaScript with optional type annotations, type-checks it and returns an optimized, minified, and human-unreadable JavaScript with stripped annotations.

In our system, we added a new mode to the compiler that reverses its standard operation: given an optimized minified JavaScript code, JSNice generates JavaScript code that is well annotated (with types) and as human-readable as possible (with useful identifier names). Our two applications for names and types were implemented as two probabilistic models that can be invoked separately.

5.1. Our dataset

To evaluate our approach, we collected two disjoint sets of JavaScript programs to form our training and evaluation data. For training, we downloaded 10,517 JavaScript projects from GitHub.[4] For evaluation, we took the 50 JavaScript projects with the highest number of commits from BitBucket.[2] By taking projects from different repositories, we decrease the likelihood of overlap between training and evaluation data.

We also searched in GitHub to check that the projects in the evaluation data are not included in the training data. Finally, we implemented a simple checker to detect and filter out minified and obfuscated files from the training and the evaluation data. After filtering minified files, we ended up with training data consisting of 324,501 files and evaluation data of 2,710 files.

5.2. Precision

To evaluate the precision of JSNice, we first minified all 2,710 files in the training data with UglifyJS[8] (other sound minifiers should produce input that is equivalent for the purposes of using JSNice). As mentioned, we focus on a particular popular form of obfuscation called layout obfuscation. It works by renaming local variables to meaningless short names and removing whitespaces and type annotations (other types of obfuscation such as encryption are not considered in this work). Each minified program is semantically equivalent (except when using `with` or `eval`) to the original. Then, we used JSNice on the minified programs to evaluate its capabilities in reconstructing name and type information. We compared the precision of the following configurations:

- The most powerful system works with all of the training data and performs structured prediction as described so far.
- Two systems using a fraction of the training data — one on 10% and one on 1% of the files.
- To evaluate the effect of structure when making predictions, we disabled relationships between unknown properties and performed predictions on that network (the training phase still uses structure).
- A naïve baseline which does no prediction: it keeps names the same and sets all types to the most common type `string`.

Name predictions. To evaluate the accuracy of name predictions, we took each of the minified programs and used the name inference in JSNice to rename its local variables. Then, we compared the new names to the original names (before obfuscation) for each of the tested programs. The results for the name reconstruction are summarized in the

Table 1. Precision and recall for name and type reconstruction of minified JavaScript programs evaluated on our test set.

System	Names accuracy (%)	Types precision (%)	Types recall (%)
all training data	63.4	81.6	66.9
10% of training data	54.5	81.4	64.8
1% of training data	41.2	77.9	62.8
all data, no structure	54.1	84.0	56.0
baseline – no predictions	25.3	37.8	100

second column of Table 1. Overall, our best system exactly recovers 63.4% of identifier names. The systems trained on less data have significantly lower precision showing the importance of training data size.

Not using structured prediction also drops the accuracy significantly and has about the same effect as an order of magnitude less data. Finally, not changing any identifier names produces accuracy of 25.3%—this is because minifying the code may not rename some variables (e.g., global variables) in order to guarantee semantic preserving transformations and occasionally one-letter local variable names stay the same (e.g., induction variable of a loop).

Type annotation predictions. Out of the 2,710 test programs, 396 have type annotations for functions in a JSDoc. For these 396, we took the minified version with no type annotations and tried to rediscover all types in the function signatures. We first ran the Closure compiler type inference, which produces no types for the function parameters. Then, we ran and evaluated JSNice on inferring these function parameter types.

JSNice does not always produce a type for each function parameter. For example, if a function has an empty body, or a parameter is not used, we often cannot relate the parameter to any known program properties and as a result, no prediction can be made and the unknown type (?) is returned. To take this effect into account, we report both recall and precision. Recall is the percentage of function parameters in the evaluation for which JSNice made a prediction other than ?. Precision refers to the percentage of cases—among the ones for which JSNice made a prediction—where it was exactly equal to the manually provided JSDoc annotation of the test programs. We note that the manual annotations are not always precise, and as a result 100% precision is not necessarily a desired outcome.

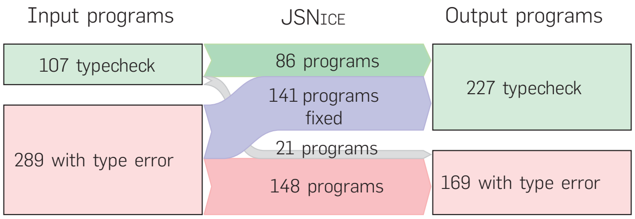
We present our evaluation results for types in the last two columns of Table 1. Since we evaluate on production JavaScript applications that typically have short methods with complex relationships, the recall for predicting program types is only 66.9% for our best system. However, we note that none of the types we infer are inferred by state-of-the-art forward type analysis (e.g., Facebook Flow[3]).

Since the total number of commonly used types is not as high as the number of names, the amount of training data has less impact on the system precision and recall. To further increase the precision and recall of type prediction, we hypothesize that adding more (semantic) relationships between program elements would be of higher importance than adding more training data. Dropping structure increases the precision of the predicted types slightly, but at the cost of a significantly reduced recall. The reason is that some types are related to known properties only transitively via other predicted types—relationships that non-structured approaches cannot capture. On the other end of the spectrum is a prediction system that suggests for every variable the most likely type in JavaScript programs—`string`. Such a system has 100% recall, but its precision is only 37.8%.

5.3. Usefulness of predicted types

To see if the predicted types are useful, we compared them to the original ones. First, we note that our evaluation data has 3,505 type annotations for function parameters in 396 programs. After removing these annotations and reconstructing them with JSNice, the number of annotations that are not ? increased to 4,114 for the same programs. The reason JSNice produces more types than originally present despite having 66.3% recall is that not all functions in the original programs had manually provided types.

Figure 7. Evaluation results for the number of type-checking programs with manually provided types and with predicted types.



Interestingly, despite annotating more functions than the original code, the output of JSNice has fewer type errors. We summarize these findings in Figure 7. For each of the 396 programs, we ran the typechecking pass of Google's Closure Compiler to discover type errors. Among others, this pass checks for incompatible types, calling into a non-function, conflicting, missing types, and non-existent properties on objects. For our evaluation, we kept all checks except the non-existent property check, which fails on almost all (even valid) programs, because it depends on annotating all properties of JavaScript classes—annotations that almost no program in training or evaluation data possesses.

When we ran typechecking on the input programs, we found the majority (289) to have typechecking errors. While surprising, this can be explained by the fact that JavaScript developers typically do not typecheck their annotations. Among others, we found the original code to have misspelled type names. Most typecheck errors occur due to missing or conflicting types. In a number of cases, the types provided were interesting for documentation, but were semantically wrong—for example, a parameter is a `string` that denotes function *name*, but the manual annotation designates its type to be `Function`. In contrast, the types reconstructed by JSNice make the majority (227) of programs typecheck. In 141 of programs that originally did not typecheck, JSNice was able to infer correct types. On the other hand, JSNice introduced type errors in 21 programs. We investigated some of these errors and found that not all of them were due to wrong types—in several cases the predicted types were rejected due to inability of the type system to precisely express the desired program properties without also manually providing type cast annotations.

5.4. Model sizes

Our models contain 7,627,484 features for names and 70,052 features for types. Each feature is stored as a triple, along with its weight. As a result we need only 20 bytes per feature, resulting in a 145.5 MB model for names and 1.3 MB model for types. The dictionary which stores all names and types requires 16.8 MB. As we do not compress our model, the memory requirements for query processing are proportional to model size.

6. CONCLUSION

We presented a new probabilistic approach for predicting program properties by learning from large codebases (termed "Big Code"). The core technical idea is to formulate the problem of property inference as structured prediction with CRFs, enabling joint predictions of program properties. As an instantiation of our method, we presented a system called JSNice that can reverse the process of layout de-obfuscation by predicting name and type annotations for JavaScript. Since its public release, JSNice has become a popular tool for JavaScript layout de-obfuscation.

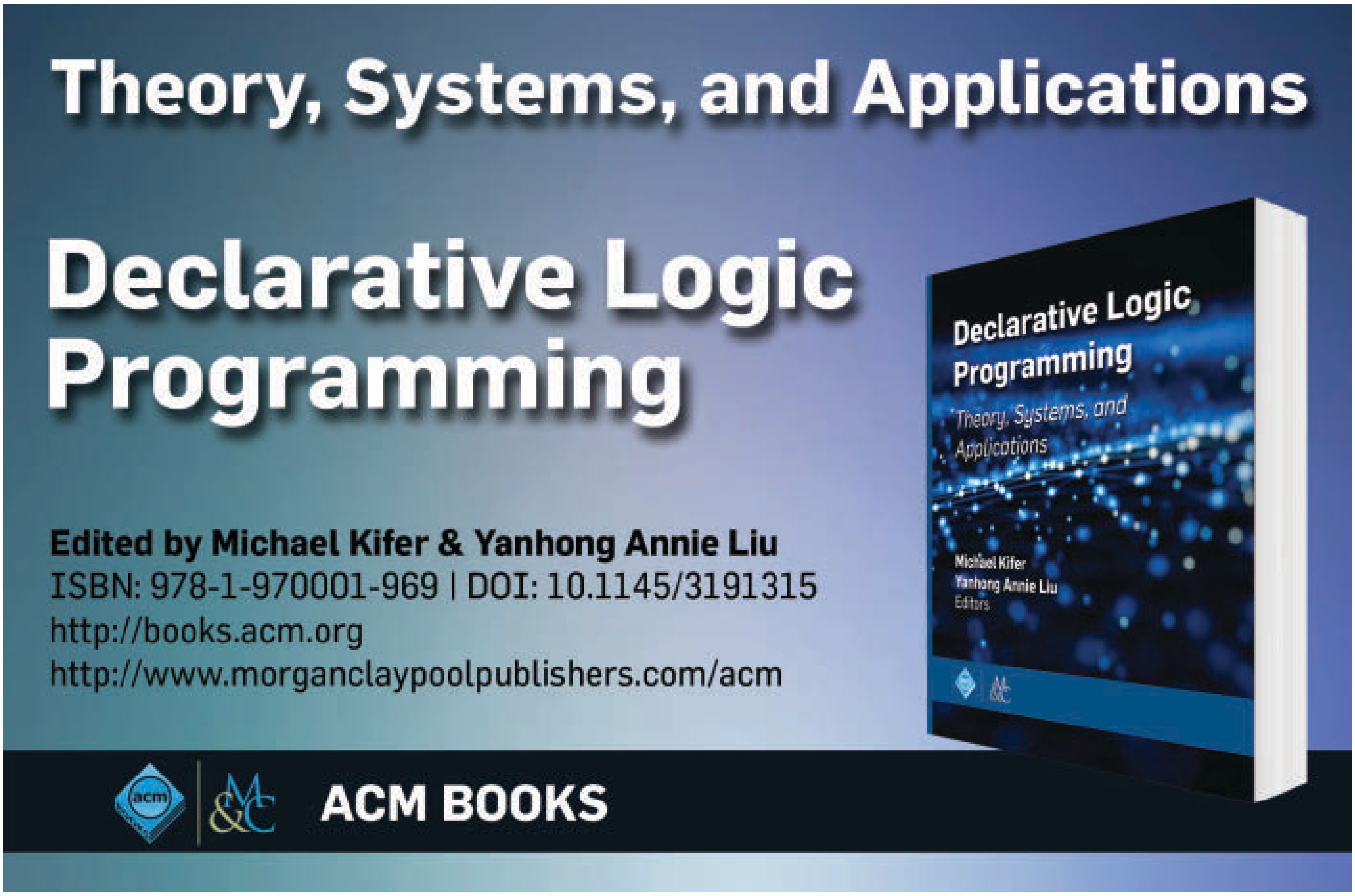
Interesting items for future work include reversing other types of obfuscation (beyond layout), extending the approach to predict semantic invariants of programs, as well as richer integration with standard program analyzers where the next prediction of the machine learning model is guided by a potential counter-example produced by the analyzer to the previous prediction.

We also remark that over the last few years the field of learning from "Big Code" has become an active area of research. Recent surveys covering various developments in this space can be found in Raychev[18] and Vechev & Yahav.[19] C

References

1. Annotating javascript. https://github.com/google/closure-compiler/wiki/Annotating-JavaScript-for-the-Closure-Compiler.
2. Bitbucket. https://bitbucket.org/.
3. Facebook flow. https://github.com/facebook/flow.
4. Github. http://github.com/.
5. Google closure compiler. https://developers.google.com/closure/compiler/.
6. Shrink your code and resources. ProGuard for Android Applications: https://developer.android.com/studio/build/shrink-code.html.
7. Typescript. https://www.typescriptlang.org/.
8. Uglifyjs. https://github.com/mishoo/UglifyJS.
9. Bichsel, B., Raychev, V., Tsankov, P., Vechev, M. Statistical deobfuscation of android applications. CCS 2016.
10. Bielik, P., Raychev, V., Vechev, M.T. PHOG: probabilistic model for code. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19–24, 2016* (2016), pp. 2933–2942.
11. DARPA. Mining and understanding software enclaves (muse). http://www.darpa.mil/news-events/2014–03–06a (2014).
12. He, X., Zemel, R.S., Carreira-Perpiñán, M.A. Multiscale conditional random fields for image labeling. *CVPR* 2004.
13. Jensen, S.H., Møller, A., Thiemann, P. Type analysis for javascript. In *Proceedings of the 16th International Symposium on Static Analysis, SAS* 2009 (Berlin, Heidelberg, 2009), Springer-Verlag, pp. 238–255.
14. Koller, D., Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, Cambridge, Massachusetts and London, England, 2009.
15. Lafferty, J.D., McCallum, A., Pereira, F.C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *ICML* 2001 (San Francisco, CA, USA, 2001), pp. 282–289.
16. Quattoni, A., Collins, M., Darrell, T. Conditional random fields for object recognition. In *NIPS* (2004), 1097–1104.
17. Ratliff, N.D., Bagnell, J.A., Zinkevich, M. (Approximate) subgradient methods for structured prediction. In *AISTATS* (2007), 380–387.
18. Raychev, V. Learning from Large Codebases. *PhD dissertation, ETH Zurich*, 2016.
19. Vechev, M., Yahav, E. Programming with "big code". *Foundations and Trends in Programming Languages 3*, 4 (2016), 231–284.

Veselin Raychev, Martin Vechev, and Andreas Krause ({veselin.raychev, martin.vechev}@inf.ethz.ch and {krausea@ethz.ch}), ETH Zurich, Zurich, Switzerland.



DOI:10.1145/3306210

To view the accompanying paper, visit doi.acm.org/10.1145/3306208

Technical Perspective
Isolating a Matching When Your Coins Go Missing

By Nisheeth K. Vishnoi

MATCHINGS IN BIPARTITE graphs play a foundational role in a variety of mathematical, scientific, and engineering disciplines—from Frobenius' work on the determinants of matrices, Gale and Shapley's influential paper on stable marriages and college admissions, Tolstoi and Kantorovich's work on the optimal transportation problem, to the online world where advertisements are being matched to users billions of times a day. As a consequence, algorithms for finding matchings in bipartite graphs also have a century-old history and the pursuit of efficient algorithms for this problem has led to major achievements in computer science and optimization.

In the 1980s, with the growing availability of parallelizable computer architectures, the study of whether one can parallelize algorithms for fundamental problems gained significant momentum. An efficient parallel algorithm would distribute the work on several processors in a manner that keeps the longest sequence of dependent tasks among processors small—ideally, logarithmic in the size of the problem. Several basic problems such as multiplying matrices, solving a system of equations and computing shortest paths in graphs already had such parallel algorithms.

For the bipartite matching problem, however, it turned out that all algorithms developed so far were inherently sequential in nature and, as such, were not amenable to parallelization. In 1985, Karp, Upfal, and Wigderson[3] presented an efficient parallel algorithm for the problem. However, there was a caveat: their algorithm was randomized, that is, it needed access to independent coin tosses. This result was soon followed by a more efficient algorithm by Mulmuley, Vazirani, and Vazirani in 1987[4] who, using an old algebraic characterization of matchings by Tutte, reduced the problem of computing matchings in graphs to computing determinants of matrices—the latter problem is known to have an efficient parallel algorithm. However, the MVV algorithm, while very different from that of Karp et al., also made crucial use of randomness in its reduction to computing determinants.

This was not the first instance of a problem in which randomness seemed to help—checking whether a number is prime or not was already a notorious example. In 1977, Solovay and Strassen[5] discovered a randomized algorithm for primality testing but no efficient algorithm that did not use randomness (called a deterministic algorithm) was discovered until 30 years later. In 1982, Valiant[6] showed that a natural routing problem on a network had an efficient randomized algorithm yet every deterministic algorithm for the problem was necessarily inefficient. Research on the power of randomness culminated in a surprising result by Kabanets and Impagliazzo in 2003[2]—removing randomness from certain efficient algorithms can show the non-existence of efficient algorithms themselves—a question that is a whisker away from the *P* versus *NP* question.

Thus, understanding the power of randomness in computation very quickly evolved from being a curiosity to being of profound interest in theoretical computer science.

Whether there exists a deterministic algorithm for primality testing or a deterministic parallel algorithm for bipartite matching remained two outstanding questions at the frontiers of our understanding of the role of randomness in computation. The former was solved in 2001 in a remarkable paper by Agrawal, Kayal and Saxena.[1] And the latter has been (nearly) recently resolved in the following paper by Fenner, Gurjar and Thierauf.

The authors show the randomized parallel algorithm of MVV can be converted into a deterministic parallel algorithm. At the heart of the MVV approach was an extremely powerful use of randomization—the Isolation Lemma. This lemma asserts there is a randomized algorithm to assign weights to the edges of a bipartite graph such that the minimum-weight perfect matching in the graph is unique—just assign to each edge a weight independently and randomly from a set of integers that is twice the number of edges in the graph. Amazingly, this randomized algorithm does not get to see the graph. Derandomizing the isolation lemma is tantamount to asking the following question—could we also find such a weight assignment when we can, in addition, no longer toss coins?

The main result in this paper is the construction of a list of weight assignments via an almost-polynomial, parallel and deterministic algorithm (which also does not look at the graph) that has the property that for any bipartite graph of a given size, at least one of the weight assignments isolates the minimum weight perfect matching in it. The end result is a gem that comes very close to solving an important open problem, makes an elegant connection between graph theory and derandomization, has been used to make progress on a few other important questions and, as a bonus, the proof is from "The Book." C

References

1. Agrawal, M., Kayal, N. and Saxena, N. PRIMES. *P. Ann. Math. 160*, 2 (2004), 781–793.
2. Kabanets, V. and Impagliazzo, R. Derandomizing polynomial identity tests means proving circuit lower bounds. In *Proceedings of the 35th Annual ACM Symp. Theory of Computing* (June 9–11, 2003, San Diego, CA, USA), 355–364.
3. Karp, R.M., Upfal, E. and Wigderson, A. Constructing a perfect matching is in random NC. In *Proceedings of the 17th Annual ACM Symp. Theory of Computing*, (May 6–8, 1985, Providence, RI, USA), 22–32.
4. Mulmuley, K., Vazirani, U.V. and Vazirani, V.V. Matching is as easy as matrix inversion. *Combinatorica 7*, 1 (1987), 105–113.
5. Solovay, R. and Strassen, V. A fast Monte-Carlo test for primality. *SIAM J. Comput. 6*, 1 (1977), 84–85.
6. Valiant, L.G. A scheme for fast parallel communication. *SIAM J. Comput. 11*, 2 (1982), 350–361.

Nisheeth K. Vishnoi is a professor of computer science at Yale University, New Haven, CT, USA.

DOI:10.1145/3306208

A Deterministic Parallel Algorithm for Bipartite Perfect Matching

By Stephen Fenner, Rohit Gurjar, and Thomas Thierauf*

Abstract

A fundamental quest in the theory of computing is to understand the power of randomness. It is not known whether every problem with an efficient randomized algorithm also has one that does not use randomness. One of the extensively studied problems under this theme is that of perfect matching. The perfect matching problem has a randomized parallel (NC) algorithm based on the Isolation Lemma of Mulmuley, Vazirani, and Vazirani. It is a long-standing open question whether this algorithm can be derandomized. In this article, we give an almost complete derandomization of the Isolation Lemma for perfect matchings in bipartite graphs. This gives us a deterministic parallel (quasi-NC) algorithm for the bipartite perfect matching problem.

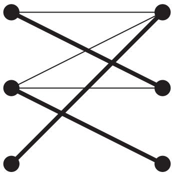
Derandomization of the Isolation Lemma means that we deterministically construct a weight assignment so that the minimum weight perfect matching is unique. We present three different ways of doing this construction with a common main idea.

1. INTRODUCTION

A perfect matching in a graph is a subset of edges such that every vertex has exactly one edge incident on it from the subset (Figure 1). The perfect matching problem, PM, asks whether a given graph contains a perfect matching. The problem has played an important role in the study of algorithms and complexity. The first polynomial-time algorithm for the problem was given by Edmonds,[7] which, in fact, motivated him to propose polynomial time as a measure of efficient computation.

Perfect matching was also one of the first problems to be studied from the perspective of parallel algorithms. A parallel algorithm is one where we allow use of polynomially many processors running in parallel. And to consider a parallel algorithm as efficient, we require the running time to be much smaller than a polynomial. In particular, the complexity class NC is defined as the set of problems which can be solved by a parallel computer with polynomially many processors in poly-logarithmic time.

Figure 1. A graph, with the bold edges showing a perfect matching.



Lovász[19] gave an efficient randomized parallel algorithm for the matching problem, putting it in the complexity class RNC (randomized NC). The essence of his parallel algorithm was a randomized reduction from the matching problem to a determinant computation. A determinant computation in turn reduces to matrix multiplication (see[4]), which is well known to have efficient parallel algorithms.

One of the central themes in the theory of computation is to understand the power of randomness, that is, whether all problems with an efficient randomized algorithm also have a deterministic one. The matching problem has been widely studied under this theme. It has been a long-standing open question whether randomness is necessary for a parallel matching algorithm, that is, whether the problem is in NC.

One can also ask for a parallel algorithm to construct a perfect matching in the graph if one exists (Search-PM). Note that there is a standard search-to-decision reduction for the matching problem, but it does not work in parallel. Karp, Upfal, and Wigderson[18] and later, Mulmuley, Vazirani, and Vazirani[21] gave RNC algorithms for Search-PM. The latter work introduced the celebrated Isolation Lemma and used it to solve Search-PM in RNC. They assign some weights to the edges of the graph, call a weight assignment *isolating* for a graph G if there is a *unique* minimum weight perfect matching in G. Here, the weight of a perfect matching is simply the sum of the weights of the edges in it. Given an isolating weight assignment with polynomially bounded integer weights, they can find the minimum weight perfect matching in G in NC (again via determinant computations).

Note that if we allow exponentially large weights then it is trivial to construct an isolating weight assignment: assign weight 2^i to the ith edge for $1 \leq i \leq m$, where m is the number of edges. This, in fact, ensures a different weight for each perfect matching. The challenge, however, is to find an isolating weight assignment with polynomially bounded weights. This is where the Isolation Lemma comes in: it states that if each edge is assigned a random weight from a polynomially bounded range then such a weight assignment is isolating with high probability.

Note that since there can be exponentially many perfect matchings in a graph, there will definitely be many collisions under a polynomially bounded weight assignment, that is, many perfect matchings will get the same weight.

The original version of this paper is entitled "Bipartite Perfect Matching is in quasi-NC" and was published in the *Proceedings of the 48th ACM Symposium on the Theory of Computing (STOC)*, 2016.

*Supported by DFG grant TH 472/4.

The beauty of the Isolation Lemma is that for the minimum weight, there will be a unique perfect matching with high probability.

Lemma 1.1 (Isolation Lemma Mulmuley, Vazirani, and Vazirani[21]). *Let $G(V, E)$ be a graph, $|E| = m$, and $w \in \{1, 2, \ldots, km\}^E$ be a uniformly random weight assignment on its edges, for some $k \geq 2$. Then w is isolating with probability at least $1 - 1/k$.*

Proof. Let e be an edge in the graph. We first give an upper bound on the probability that there are two minimum-weight perfect matchings, one containing e and other not containing e. For this, say the weight of every other edge except e has been fixed. Let W be the minimum weight of any perfect matching that avoids e, and let $W' + w(e)$ be the minimum weight of any perfect matching that contains e.

Now, what is the probability that these two minimum weights are equal? Since W and W' are already fixed by the other edges, and $w(e)$ is chosen uniformly and randomly between 1 and km,

$$\Pr\{W = W' + w(e)\} \leq \frac{1}{km}.$$

By the union bound,

$$\Pr\{\exists e \in E \; W = W' + w(e)\} \leq \frac{m}{km} = \frac{1}{k}.$$

Now, to finish the proof, observe that there is a unique minimum weight perfect matching if and only if there is no such edge with the above property. □

One way to obtain a deterministic parallel (NC) algorithm for the perfect matching problem is to derandomize this lemma. That is, to *deterministically* construct such a polynomially bounded isolating weight assignment in NC. This has remained a challenging open question.

Derandomization of the Isolation Lemma has been known for some special classes of graphs, for example, planar bipartite graphs,[6,26] strongly chordal graphs,[5] and graphs with a small number of perfect matchings.[1,14] Here, we present an almost complete derandomization of the Isolation Lemma for bipartite graphs. The class of bipartite graphs appears very naturally in the study of perfect matchings. A graph is bipartite if there is a partition of its vertex set into two parts such that each edge connects a vertex from one part to a vertex from the other (the graph in Figure 1 is bipartite). Thus, a perfect matching in a bipartite graph matches every vertex in one part to exactly one vertex in the other.

In Section 3, we construct an isolating weight assignment for bipartite graphs with quasi-polynomially large ($n^{O(\log n)}$) weights, where n is the number of vertices in the graph. Note that this is slightly worse than what we would have ideally liked, which is—polynomially bounded weights. Hence, we do not get an NC algorithm. Instead, we get that for bipartite graphs, the problems PM and Search-PM are in quasi-NC^2. That is, the problems can be solved in $O(\log^2 n)$ time using $n^{O(\log n)}$ parallel processors. A more detailed exposition is in the conference version of the article.[10]

Theorem 1.2. *For bipartite graphs,* PM *and* Search-PM *are in quasi-*NC^2.

1.1. The isolation technique

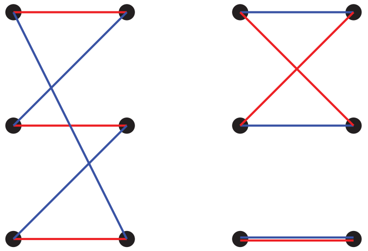
At the heart of our isolation approach is a cycle elimination technique. It is easy to see that if we take a union of two perfect matchings, we get a set of disjoint cycles and singleton edges (see Figure 2). Each of these cycles has even length and has edges alternating from the two perfect matchings. Cycles thus play an important role in isolating a perfect matching. Given a weight assignment on the edges, let us define the *circulation* of an even cycle C to be the difference of weights between the set of odd-numbered edges and the set of even-numbered edges in cyclic order around C. Clearly, if all the cycles in the union of two perfect matchings have zero circulations, then the two perfect matchings will have the same weight. It turns out that the converse is also true when the two perfect matchings under consideration are of the minimum weight.[6] This observation is the starting point of our cycle elimination technique.

In the case of bipartite graphs, this observation can be further generalized. We show that for any weight assignment w on the edges of a bipartite graph, if we consider the union of all the minimum weight perfect matchings, then it has only those cycles which have zero circulation (Lemma 2.1). This means that if we design the weights w such that a particular cycle C has a *nonzero* circulation, then C does not appear in the union of minimum weight perfect matchings, that is, at least one of the edges in C does not participate in any minimum weight perfect matchings. This is the way we will be eliminating cycles.

If we eliminate all cycles this way, we will get a unique minimum weight perfect matching, for if there were two minimum weight perfect matchings, their union would contain a cycle. However, it turns out that there are too many cycles in the graph, and it is not possible to ensure nonzero circulations simultaneously for all cycles while keeping the edge weights small (proved in[17]). Instead, what is achievable is nonzero circulation for any *polynomially large* set of cycles using well-known hashing techniques. In short, we can eliminate any desired set of a small number of cycles at once. With this tool in hand we would like to eliminate all cycles—whose number can be exponentially large—in a small number of rounds.

We present three different ways of achieving this. The first two of these have appeared before in different versions of our article.[10] The third has not appeared anywhere before.

Figure 2. Two perfect matchings, with red and blue edges, respectively. Their union forms a set of disjoint cycles and edges.



1. In the first approach, in the ith round, we eliminate all cycles of length at most 2^{i+1}. Hence, we eliminate all cycles in log n rounds. Each round is efficient because if a graph does not have any cycles of length at most ℓ, then the number of cycles up to length 2ℓ is polynomially bounded.[25, 23]
2. In the second approach, first we eliminate all cycles of length at most 4 log n. The bound we have on the number of such cycles is quasi-polynomial in n. Alon, Hoory, and Linial[2] have shown that any graph which does not contain any cycle of length $\leq 4 \log n$ must have average degree at most 2.5, and thus must have at least a constant fraction of nodes with degree 2 or less. From the resulting graph, we remove all nodes of degree 1, and we contract degree-2 nodes one by one (identifying the two neighbors), until there are no degree-2 nodes left. This creates new small cycles in the graph. We then repeat the procedure of eliminating cycles of length at most 4 log n from the new graph. In each round the number of nodes decreases by a constant fraction. Thus, after $O(\log n)$ rounds, we eliminate all nodes and hence, all cycles.
3. In the third approach, instead of considering the lengths of the cycles, we try to pick as many edge-disjoint cycles as possible and eliminate them. Note that edge-disjointness ensures that we will eliminate at least as many edges as cycles. Erdös and Pósa[9] showed that any graph with m edges and n nodes contains $\Omega\left(\frac{m-n}{\log(m-n)}\right)$ edge-disjoint cycles. A careful argument shows that in $O(\log^2 n)$ rounds, we eliminate enough edges so that no cycles are left.

As we will see later, the first approach is more efficient than the other two. We still think it is interesting to see different ways of achieving isolation, as they might lead to better ideas for getting isolation with polynomially bounded weights or isolation in other settings. Another interesting point is that our second approach was used in designing a *pseudo-deterministic* RNC algorithm for bipartite matching.[13]

Our crucial technical result (Lemma 2.1) about eliminating cycles has a proof based on linear programming (LP) duality. In the next section, we describe a LP formulation for bipartite perfect matching and its dual, and then use it to prove our result. Finally in Section 3, we formally describe the weight construction and the three approaches to eliminate all cycles.

2. CYCLE ELIMINATION VIA NONZERO CIRCULATIONS

In this section, we formally describe our main technical tool which enables cycle elimination. Let us first give a formal definition of cycle circulation. For a weight assignment w on the edges of a graph G, the *circulation* $circ_w(C)$ of an even-length cycle $C = (v_1, v_2, \ldots, v_k)$ is defined as the alternating sum of the edge weights around C:

$$circ_w(C) = \left| w(v_1,v_2) - w(v_2,v_3) + w(v_3,v_4) - \cdots - w(v_k,v_1) \right|.$$

The definition is independent of the edge we start with because we take the absolute value of the alternating sum.

The following lemma about circulations of cycles gives us a way to eliminate cycles. For a weight assignment w on the edges of a graph G, let G_w be the union of minimum weight perfect matchings, that is, it is a subgraph of G that has exactly those edges that are present in some minimum weight perfect matching in G.

LEMMA 2.1 (ZERO CIRCULATION). *Let w be a weight function on the edges of a bipartite graph G. Let C be a cycle in the subgraph G_w. Then $circ_w(C) = 0$.*

The following corollary is immediate, which shows how the above lemma can be used to eliminate cycles.

COROLLARY 2.2 (CYCLE ELIMINATION). *Let C be a cycle in a bipartite graph G and w be a weight function on its edges such that $circ_w(C) \neq 0$. Then C is not present in G_w.*

There are several ways to prove Lemma 2.1.

1. In our original article,[10] we presented a proof based on properties of the *perfect matching polytope*. In the argument, the center point of the polytope is slightly moved along cycle C, so that the point stays in the polytope. This implies that the circulation of C must be zero.
2. After the first version of our article was published, Rao, Shpilka, and Wigderson (see [Goldwasser and Grossman,[13] Lemma 2.4]) came up with an alternate proof of Lemma 2.1, similar to ours, but based on *Hall's Theorem* instead of the matching polytope.
3. In a column for SIGACT News,[11] we gave a *geometric proof*, where we just argue via vectors being parallel or perpendicular to each other. One might consider this the shortest and most elegant of the proofs.
4. Nevertheless, in Section 2.1 below, we present a fourth proof that we find very nice. It is based on *LP duality*.

2.1. LP formulation for perfect matching

The minimum weight perfect matching problem on bipartite graphs has a simple and well-known LP formulation. Let G be a bipartite graph with vertex set V and edge set E. Then the following linear program captures the minimum weight perfect matching problem (see, for example, Lovász and Plummer[20]).

$$\min \sum_{e \in E} w_e x_e$$

$$x_e \geq 0 \quad \forall e \in E \tag{1}$$

$$\sum_{e \in \delta(v)} x_e = 1 \quad \forall v \in V, \tag{2}$$

where $\delta(v)$ denotes the set of edges incident on a vertex v. The linear program has one variable x_e for each edge in the graph. Intuitively, $x_e = 1$ represents that the edge e is present in the perfect matching and $x_e = 0$ represents that e is not in the perfect matching. Then, Equation (2) is simply saying that a perfect matching contains exactly one edge from the set of edges incident on a particular vertex. The objective function asks to minimize the sum of the weights of the edges present in a perfect matching, that is, the weight of a perfect matching.

An important point to note here is that the above LP formulation works only for bipartite graphs, and this is the reason our proof does not work for general graphs.

From the standard theory of LP duality, the following is the dual linear program for minimum weight perfect matching. Since in the primal LP above we had an equality constraint for each vertex, here we have a variable for each vertex.

$$\max \sum_{u \in V} y_u$$
$$y_u + y_v \le w_e \qquad \forall e = (u, v) \in E. \tag{3}$$

Note that the dual variables do not have any non-negativity constraint, since the primal constraints are equalities.

It follows from strong LP duality that the optimal values of these two linear programs are equal. This will be crucial for the proof of Lemma 2.1.

Proof of Lemma 2.1. Let $e = (u, v)$ be any edge participating in a minimum weight perfect matching, in other words, e is an edge in G_w. Let $y \in {}^V$ be a dual optimal solution. We claim that the dual constraint (3) corresponding to e is tight, that is,

$$y_u + y_v = w_e. \tag{4}$$

This can be seen as follows: for any minimum weight perfect matching M, its weight by definition is the primal optimal value, and thus, by strong duality must be equal to the dual optimal value. That is,

$$w(M) = \sum_{v \in V} y_v.$$

Note that a sum over all vertices is the same as a sum over the end points of all the edges in a perfect matching. Thus,

$$\sum_{e \in M} w(e) = \sum_{e=(u,v) \in M} (y_u + y_v). \tag{5}$$

Together with (3), Equation (5) implies Equation (4).

Now, let $C = (u_1, u_2, \ldots, u_{2k})$ be a cycle in G_w. Since each edge in C is part of some minimum weight perfect matching, by (4), all the edges of C are tight w.r.t. the dual optimal solution y. Hence,

$$\begin{aligned} circ_w(C) &= w(u_1, u_2) - w(u_2, u_3) + \cdots - w(u_{2k}, u_1) \\ &= (y_{u_1} + y_{u_2}) - (y_{u_2} + y_{u_3}) + \cdots - (y_{u_{2k}} + y_{u_1}) \\ &= 0. \end{aligned}$$

Hence, every cycle in G_w has zero circulation. □

3. CONSTRUCTING AN ISOLATING WEIGHT ASSIGNMENT

Corollary 2.2 gives us a way to eliminate cycles. Suppose C is a cycle in graph G. If we construct a weight assignment w such that $circ_w(C) \neq 0$ then the cycle C will not be present in G_w, that is, at least one edge of C will be missing.

We will be applying this technique on a small set of chosen cycles. As mentioned earlier, there are standard ways to construct a weight function which ensures nonzero circulations for any small set of cycles simultaneously, see for example.[12]

Lemma 3.1. *Let $\mathcal{C}$ be any set of s cycles in graph $G(V, E)$ and let $E = \{e_1, e_2, \ldots, e_m\}$. For $j \in$, we define weights*

$$w_{(\text{mod } j)}(e_i) := 2^{i-1} \bmod j, \quad \textit{for } i = 1, 2, \ldots, m.$$

Then there exists a $j \le ms$ such that

$$circ_{w_{(\text{mod } j)}}(C) \neq 0, \quad \textit{for all } C \in \mathcal{C}.$$

Note that the above lemma actually gives a list of weight functions such that for any desired set of cycles, at least one of the weight functions in the list works. Also observe that the weight of any edge under any of these functions is bounded by ms. That is, the weights are polynomially bounded as long as the number of cycles is.

The isolating weight assignment is now constructed in rounds. The strategy is to keep eliminating a small number (poly or quasi-poly) of cycles in each round by giving them nonzero circulations. This is repeated until we are left with no cycles. In every round, we add the new weight function to the current weight function on a smaller scale. This is to ensure that the new weights do not interfere significantly with the circulations of cycles which have been already eliminated in earlier rounds.

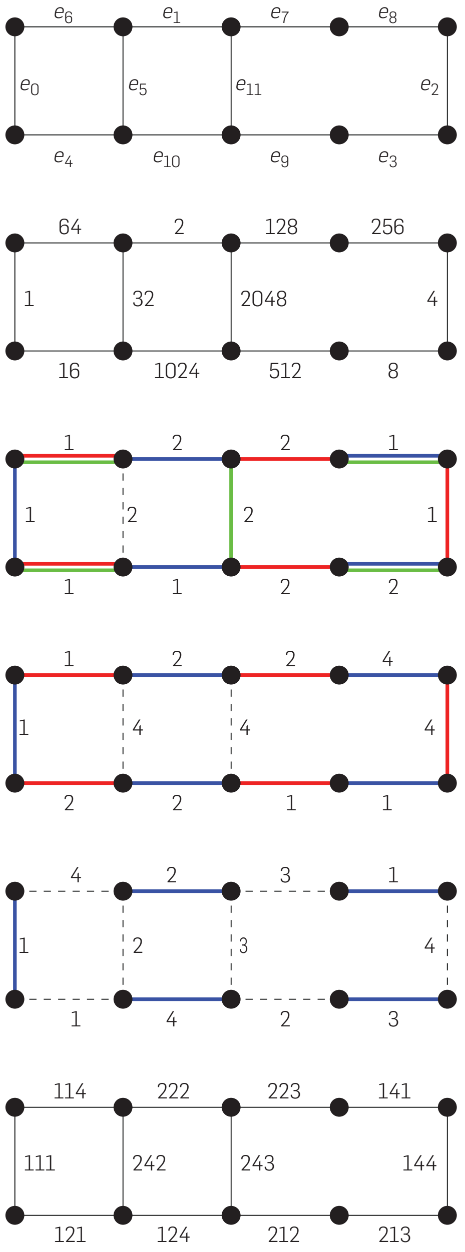
In more detail, if w_i is the current weight function in the ith round, then in the next round, we will consider the weight function $w_{i+1} = Nw_i + w'$, for some weight function w' and a large enough number N. The number N is chosen to be larger than $n \cdot \max_e w'(e)$, which ensures that Nw_i gets precedence over w'. The weight function w' is designed to ensure nonzero circulations for a desired set of cycles in G_{w_i}. These cycles will not appear in $G_{w_{i+1}}$. We will keep eliminating cycles in this way until we obtain a w such that G_w has no cycles. Recall that G_w is defined to be the union of minimum weight perfect matchings with respect to w, and thus, contains at least one perfect matching. Since G_w has no cycles, it must have a unique perfect matching, and so, w is isolating for G. Figure 3 shows a graph where an isolating weight assignment is constructed in 3 rounds using our Approach 1, described below.

Bound on the weights. If we want to assign nonzero circulations to at most s cycles in each round, then the weights are bounded by ms by Lemma 3.1. If there are k such rounds, the bound on the weights becomes $O((nms)^k)$. As we will see later, the quantity $(nms)^k$ will be quasi-polynomially bounded.

Recall that Lemma 3.1 gives a list of ms candidate weight functions such that at least one of them gives nonzero circulations to all the s cycles chosen in a round. We need to try all possible $(ms)^k$ combinations of these candidate functions coming from each round. Our quasi-NC algorithm tries all these combinations in parallel.

Now, the crucial question left in our isolating weight construction is this: how to eliminate all cycles, which are

Figure 3. Iterative construction of an isolating weight assignment on a bipartite graph.



G is a 10-node graph with 12 edges $e_0, \dots, e_{11}$.

The initial weights are $w(e_i) := 2^i$.

Take $w_{(\text{mod } 3)}$. The 4-cycles are gone, but only e_5 is removed in the derived graph—the union of 3 min matchings (blue, red, green).

Take $w_{(\text{mod } 7)}$. The 6-cycles are gone, but only e_{11} is removed in the derived graph—the union of the blue and red matchings (the non-min-weight green does not survive).

Take $w_{(\text{mod } 5)}$. The 10-cycle is now gone and only the blue matching survives in the derived graph.

Combining the reduced weights gives us a weight function that isolates the blue matching as unique with min weight. Numbers can be interpreted in any radix ≥5 in this example.

possibly exponentially many, in a small number of rounds, while only eliminating a small number of cycles in each round. We present three different approaches for this. Each approach will have a different criterion for choosing a small set of cycles, which are to be eliminated in a round. The rest of the procedure is common to all three approaches. The following table gives, for each approach, the number of cycles chosen in each round and the number of rounds required to eliminate all cycles. Here we use $m \leq n^2$.

	Number of cycles in each round	Number of rounds	Bound on the weights
Approach 1	n^4	$O(\log n)$	$n^{O(\log n)}$
Approach 2	$n^{O(\log n)}$	$O(\log n)$	$n^{O(\log^2 n)}$
Approach 3	$O(n^2)$	$O(\log^2 n)$	$n^{O(\log^2 n)}$

3.1. Approach 1: Doubling the lengths of the cycles

Here, the idea is to double the length of the cycles that we want to eliminate in each round. There will be $\log n$ rounds. In the ith round, we eliminate all cycles of length at most 2^{i+1}, and thus we eliminate all cycles in $\log n$ rounds. The following lemma shows that if we have already eliminated all the cycles of length at most 2^i then the number of cycles of length 2^{i+1} is polynomially bounded, for any i.

LEMMA 3.2 (Subramanian[23]). *Let H be a graph with n nodes that has no cycles of length at most r, for some even number $r \geq 4$. Then H has at most n^4 cycles of length at most $2r$.*

PROOF. Let C be a cycle of length $\leq 2r$ in G. We choose four vertices u_0, u_1, u_2, u_3 on C, which divide it into four almost equal parts. We *associate* the tuple (u_0, u_1, u_2, u_3) with C. We claim that C is the only cycle associated with (u_0, u_1, u_2, u_3). For the sake of contradiction, let there be another such cycle C'. Let $p \neq p'$ be paths of C and C', respectively, that connect the same u-nodes. As the four segments of C and C' are of equal length, we have $|p|, |p'| \leq r/2$. Thus, p and p' create a cycle of length $\leq r$, which is a contradiction. Hence, a tuple is associated with only one cycle. The number of tuples of four nodes is bounded by n^4 and so is the number of cycles of length $\leq 2r$. □

3.2. Approach 2: Eliminating small cycles implies a small average degree

Here, the idea is to use a result of Alon, Hoory, and Linial,[2] which states that a graph with no small cycles must have many nodes of degree ≤ 2. To get an intuitive understanding of this, consider a graph where each node has degree at least 3: do a breadth-first search of the graph starting from an arbitrary node until depth log n. When one reaches a node v via an edge e, there are at least 2 edges incident on v other than e. So, the search tree contains a binary tree of depth log n. The nodes in the tree cannot be all distinct, because otherwise we would have strictly more than $2^{\log n} = n$ nodes. A node that appears twice in the search tree gives us a cycle of length at most 2 log n. In other words, if there are no cycles of length at most 2 log n, then the graph must have a node with degree 2 or less. Alon, Hoory, and Linial[2] generalize this intuition to show that as the length of the shortest cycle increases, the average degree gets closer to 2.

Lemma 3.3 (Alon, Hoory, and Linial). *Let H be a graph with no cycles of length $<4 \log n$. Then H has average degree <2.5.*

In this approach, we start by eliminating all cycles in graph G of length $\leq 4 \log n$. It is easy to see that the number of such cycles will be bounded by $n^{4 \log n}$. Lemma 3.3 implies that after this, a constant fraction of the nodes in G have degree ≤ 2.

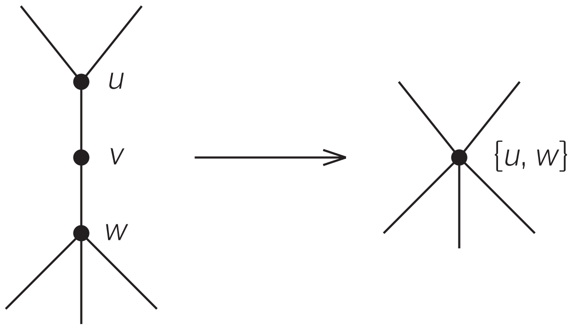
Having many nodes of degree ≤ 2 is quite useful when we are interested in perfect matchings because they provide a way to shrink the graph while preserving perfect matchings.

1. Let v be node of degree 1 in G and u be the unique neighbor of v. Recall that our graph after every round is always a union of perfect matchings. Therefore, u has degree 1 as well. Hence, we can simply delete u and v from G.
2. Let v be a node of degree 2 in G with its neighbors u and w. Now, construct a new graph G' from G by deleting v and identifying u and w to get a single node $\{u, w\}$ (see Figure 4). We refer to this operation as *collapsing* the node v. Observe that perfect matchings of G and G' are in one-to-one correspondence.

Note also that any cycle in G appears in G' with the degree-2 nodes cut out, that is, cycles get shorter in G'.

To further proceed in this approach, we first collapse all degree-2 nodes in G (one by one) and delete all degree-1 nodes. Let G_0 be the resulting graph. Since there were a constant fraction of nodes with degree ≤ 2 in G, the number of nodes in G_0 decreases by a constant fraction. Note also that all nodes in G_0 have degree ≥ 3. By Lemma 3.3, graph G_0 again has small cycles of length $\leq 4 \log n$. Now, we can repeat the procedure of eliminating all cycles of length $\leq 4 \log n$ with G_0.

Figure 4. Deleting a degree-2 node *v* and identifying its two neighbors *u* and *w*—an operation which preserves perfect matchings and cycles.



In every round, the number of nodes in the graph decreases by a constant fraction. Thus, in $O(\log n)$ rounds, we eliminate all cycles and reach the empty graph. One can easily obtain a unique perfect matching in the original graph G, by reversing all the degree-1 deletions and degree-2 collapses.

3.3. Approach 3: Eliminating a maximum size set of edge-disjoint cycles

In this approach, we do not consider the lengths of the cycles. Instead, in each round we pick as many edge-disjoint cycles as possible. Recall that eliminating a cycle means that at least one of its edges will not be present in the graph in the next round. Hence, when we eliminate a set of edge-disjoint cycles, we will eliminate at least as many edges. Once we remove enough edges, we will be left with no cycles.

Let G be a graph with n vertices and m edges. The number of cycles picked in each round is trivially bounded by m. The non-trivial part is to come up with a lower bound. Erdös and Pósa[9] showed that G has at least $\frac{m-n}{O(\log(m-n))}$ edge-disjoint cycles. We will argue that if we eliminate a maximum size set of edge-disjoint cycles in a round, then the quantity $m - n$ decreases by a significant fraction in every round.

Lemma 3.4. *Let G be a connected graph with n vertices and m edges. Let $\mathcal{C}$ be a maximum size set of edge-disjoint cycles in G. Let H be any subgraph of G obtained by deleting at least one edge from each cycle in $\mathcal{C}$. Then for any connected component H_1 of H with n_1 vertices and m_1 edges,*

$$m_1 - n_1 \leq (m-n)\left(1 - \frac{1}{O(\log(m-n))}\right).$$

Proof. Let $|\mathcal{C}| = k$. Let H' be any subgraph of G such that H is a subgraph of H' and for each cycle in $\mathcal{C}$, *exactly* one edge is missing in H'. Note that H' is still connected, since the cycles in $\mathcal{C}$ are edge-disjoint. The difference between the number of edges and vertices of H' is $m - n - k$.

Since H is obtained by deleting possibly some more edges from H', for any connected component of H, the difference between the number of edges and vertices cannot be larger than $m - n - k$. Now, the lemma follows from the above lower bound of Erdös and Pósa[9] on the number of edge-disjoint cycles. □

Let us repeat the procedure of eliminating a maximum size set of edge-disjoint cycles. It follows from the lemma that after $O(\log^2 n)$ rounds, each component of the obtained graph will have a constant difference between the number of edges and vertices. At this stage, each component will have only constantly many cycles. And so, in one more round we will eliminate all cycles.

A different view on the third approach is by considering the *dimension* of the perfect matching polytope. For a connected bipartite graph, where each of its edges belong to some perfect matching, the perfect matching polytope has dimension $m - n + 1$ [Lovász and Plummer,[20] Theorem 7.6.2]. Thus, the argument of this approach can also be viewed as decreasing the dimension of the perfect matching polytope by a fraction in each round and eventually reaching dimension zero, that is, just one perfect matching point.

4. FURTHER DEVELOPMENTS

After years of inactivity, our result inspired a series of follow-up works on parallel algorithms for perfect matching and the Isolation Lemma. In one direction, our isolation approach was generalized to the broader settings of matroid intersection and polytopes with totally unimodular faces, respectively.[15,16] For these general settings, the right substitute for cycles are integer vectors parallel to a face of the associated polytope. Following our first approach, if one eliminates vectors of length $\leq 2^i$, then there are only polynomially many vectors of length $\leq 2^{i+1}$, in their respective settings (see[15,16] for details). It is not clear, however, if our second and third approaches work in these settings.

In another direction, Svensson and Tarnawski[24] generalized the isolation result to perfect matchings in *general graphs*. They use the basic framework of our first approach as the starting point, but they need to combine the technique of eliminating cycles with a second parameter (*contractability*) to control the progress in subsequent rounds.

The techniques developed by us and by Svensson and Tarnawski were used by Anari and Vazirani[3] to compute a perfect matching in *planar graphs* in NC (see also[22]), which also was a long-standing open problem. They show that the sets to be contracted, odd sets of vertices that form a tight cut in the LP-constraints, can be computed in NC. In a subsequent work, the NC algorithm was further generalized to one-crossing-minor-free graphs.[8]

The Isolation Lemma has applications in many different settings—in particular, in design of randomized algorithms. The main open question that remains is for what other settings can one derandomize the Isolation Lemma. We conjecture that our isolation approach works for any family of sets whose corresponding polytopes are described by 0/1 constraints.

Acknowledgments

We would like to thank Manindra Agrawal and Nitin Saxena for their constant encouragement and very helpful discussions. We thank Arpita Korwar for discussions on some other techniques used in this research, Jacobo Torán for discussions on the number of shortest cycles, and Nisheeth Vishnoi for helpful comments.

References

1. Agrawal, M., Hoang, T.M., Thierauf, T. The polynomially bounded perfect matching problem is in NC^2. In *24th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 4393 of *Lecture Notes in Computer Science* (Berlin Heidelberg, 2007). Springer, 489–499.
2. Alon, N., Hoory, S., Linial, N. The Moore bound for irregular graphs. *Graphs Comb. 18*, 1 (2002), 53–57.
3. Anari, N., Vazirani, V.V. Planar graph perfect matching is in NC. Technical Report arXiv:1709.07822, CoRR, 2017.
4. Berkowitz, S.J. On computing the determinant in small parallel time using a small number of processors. *Inform. Process. Lett. 18*, 3 (1984), 147–150.
5. Dahlhaus, E., Karpinski, M. Matching and multidimensional matching in chordal and strongly chordal graphs. *Discrete Appl. Math. 84* (1998), 79–91.
6. Datta, S., Kulkarni, R., Roy, S. Deterministically isolating a perfect matching in bipartite planar graphs. *Theory Comput. Syst. 47* (2010), 737–757.
7. Edmonds, J. Paths, trees, and owers. *Can. J. Math. 17* (1965), 449–467.
8. Eppstein, D., Vazirani, V.V. NC algorithms for perfect matching and maximum flow in one-crossing-minor-free graphs. Technical Report arXiv:1802.00084, CoRR, 2018.
9. Erdös, P., Pósa, L. On the maximal number of disjoint circuits of a graph. *Publ. Math. Debrecen 9* (1962), 3–12.
10. Fenner, S., Gurjar, R., Thierauf, T. Bipartite Perfect Matching is in quasi-NC. In *Proceedings of the 48th ACM Symposium on the Theory of Computing (STOC)* (Cambridge, MA, USA, June 18–21, 2016). arXiv:1601.06319; ECCC TR15-177.
11. Fenner, S., Gurjar, R., Thierauf, T. Guest column: Parallel algorithms for perfect matching. *SIGACT News 48*, 1 (Mar. 2017), 102–109.
12. Fredman, M.L., Komlós, J., Szemerédi, E. Storing a sparse table with $O(1)$ worst case access time. *J. ACM*, *31*, 3 (June 1984), 538–544.
13. Goldwasser, S., Grossman, O. Bipartite perfect matching in pseudo-deterministic NC. In *44th International Colloquium on Automata, Languages, and Programming* (Warsaw, Poland, July 10–14, 2017), ICALP 2017, 13, 87:1–87.
14. Grigoriev, D., Karpinski, M. The matching problem for bipartite graphs with polynomially bounded permanents is in NC (extended abstract). In *28th Annual Symposium on Foundations of Computer Science (FOCS)* (Los Angeles, California, USA, October 27–29, 1987), 166–172.
15. Gurjar, R., Thierauf, T. Linear matroid intersection is in quasi-NC. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing* (Montreal, QC, Canada, June 19–23, 2017), STOC 2017, 821–830.
16. Gurjar, R., Thierauf, T., Vishnoi, N.K. Isolating a vertex via lattices: Polytopes with totally unimodular faces. In *45th International Colloquium on Automata, Languages, and Programming (ICALP)* (Prague, Czech Republic, July 9–13, 2018), 74:1–74:14.
17. Kane, D., Lovett, S., Rao, S. The independence number of the birkhoff polytope graph, and applications to maximally recoverable codes. In *58th IEEE Annual Symposium on Foundations of Computer Science* (Berkeley, CA, USA, October 15–17, 2017), FOCS 2017, 252–259.
18. Karp, R.M., Upfal, E., Wigderson, A. Constructing a perfect matching is in random NC. *Combinatorica*, *6*, 1 (1986), 35–48.
19. Lovász, L. On determinants, matchings, and random algorithms. In *Fundamentals of Computation Theory* (Berlin/Wendisch-Rietz (GDR), September 17–21, 1979), 565–574.
20. Lovász, L., Plummer, M.D. *Matching Theory. North-Holland mathematics studies*. Elsevier Science Ltd, 1986.
21. Mulmuley, K., Vazirani, U.V., Vazirani, V.V. Matching is as easy as matrix inversion. *Combinatorica*, *7* (1987), 105–113.
22. Sankowski, P. NC algorithms for weighted planar perfect matching and related problems. In *45th International Colloquium on Automata, Languages, and Programming (ICALP)* (Prague, Czech Republic, July 9–13, 2018), 97:1–97:16.
23. Subramanian, A. A polynomial bound on the number of light cycles in an undirected graph. *Inform. Process. Lett. 53*, 4 (1995), 173–176.
24. Svensson, O., Tarnawski, J. The matching problem in general graphs is in quasi-NC. In *58th IEEE Annual Symposium on Foundations of Computer Science* (Berkeley, CA, USA, October 15–17, 2017), FOCS 2017, 696–707.
25. Teo, C.P., Koh, K.M. The number of shortest cycles and the chromatic uniqueness of a graph. *J. Graph Theory 16*, 1 (1992), 7–15.
26. Tewari, R., Vinodchandran, N. Green's theorem and isolation in planar graphs. *Inform. Comput. 215* (2012), 1–7.

Stephen Fenner (fenner.sa@gmail.com), University of South Carolina, Columbia, SC, USA.

Rohit Gurjar (rohitgurjar0@gmail.com), California Institute of Technology, Pasadena, CA, USA.

Thomas Thierauf (thomas.thierauf@uniulm.de), Aalen University, Germany.

CAREERS

National University of Singapore

Sung Kah Kay Assistant Professorship in all areas of Computer Science

The Department of Computer Science at the National University of Singapore (NUS) invites applications for the Sung Kah Kay Assistant Professorship. Applicants can be in any area of computer science. This prestigious chair was set up in memory of the late Assistant Professor Sung Kah Kay. Candidates should be early in their academic careers and yet demonstrate outstanding research potential, and strong commitment to teaching.

The Department enjoys ample research funding, moderate teaching loads, excellent facilities, and extensive international collaborations. We have a full range of faculty covering all major research areas in computer science and boasts a thriving PhD program that attracts the brightest students from the region and beyond. More information is available at www.comp.nus.edu.sg/careers.

NUS is an equal opportunity employer that offers highly competitive salaries, and is situated in Singapore, an English-speaking cosmopolitan city that is a melting pot of many cultures, both the east and the west. Singapore offers high-quality education and healthcare at all levels, as well as very low tax rates.

Application Details:

- ► Submit the following documents (in a single PDF) online via: https://faces.comp.nus.edu.sg.
  - A cover letter that indicates the position applied for and the main research interests
  - Curriculum Vitae
  - A teaching statement
  - A research statement
- ► Provide the contact information of 3 referees when submitting your online application, or, arrange for at least 3 references to be sent directly to csrec@comp.nus.edu.sg.
- ► Application reviews will commence immediately and continue until the position is filled
- ► If you have further enquiries, please contact the Search Committee Chair, Weng-Fai Wong, at csrec@comp.nus.edu.sg.

Southern University of Science and Technology (SUSTech)

Professor Position in Computer Science and Engineering

The Department of Computer Science and Engineering (CSE, http://cse.sustc.edu.cn/en/), Southern University of Science and Technology (SUSTech) has multiple Tenure-track faculty openings at all ranks, including Professor/Associate Professor/Assistant Professor. We are looking for outstanding candidates with demonstrated research achievements and keen interest in teaching, in the following areas (but are not restricted to):

- ► Data Science
- ► Artificial Intelligence
- ► Computer Systems (including Networks, Cloud Computing, IoT, Software Engineering, etc.)
- ► Cognitive Robotics and Autonomous Systems
- ► Cybersecurity (including Cryptography)

Applicants should have an earned Ph.D. degree and demonstrated achievements in both research and teaching. The teaching language at SUSTech is bilingual, either English or Putonghua. It is perfectly acceptable to use English in all lectures, assignments, exams. In fact, our existing faculty members include several non-Chinese speaking professors.

As a State-level innovative city, Shenzhen has identified innovation as the key strategy for its development. It is home to some of China's most successful high-tech companies, such as Huawei and Tencent. SUSTech considers entrepreneurship as one of the main directions of the university. Strong supports will be provided to possible new initiatives. SUSTech encourages candidates with experience in entrepreneurship to apply.

The Department of Computer Science and Engineering at SUSTech was founded in 2016. It has 17 professors, all of whom hold doctoral degrees or have years of experience in overseas universities. Among them, three are IEEE fellows; one IET fellow. The department is expected to grow to 50 tenure track faculty members eventually, in addition to teaching-only professors and research-only professors.

SUSTech is committed to increase the diversity of its faculty, and has a range of family-friendly policies in place. The university offers competitive salaries and fringe benefits including medical insurance, retirement and housing subsidy, which are among the best in China. Salary and rank will commensurate with qualifications and experience. More information can be found at http://talent.sustc.edu.cn/en.

We provide some of the best start-up packages in the sector to our faculty members, including one PhD studentship per year, in addition to a significant amount of start-up funding (which can be used to fund additional PhD students and postdocs, research travels, and research equipments).

To apply, please provide a cover letter identifying the primary area of research, curriculum vitae, and research and teaching statements, and forward them to cshire@sustc.edu.cn.

University of Alabama

Programmer Analysts III (6)

The University of Alabama seeks 6 Programmer Analysts III for its Tuscaloosa, AL Location.

- ► Master's degree; OR Bachelor's degree and two (2) years of IT experience. Must include some .NET C# experience.
- ► Ability to effectively communicate, both verbally and in writing with uses of varying degrees of technical ability and understanding.
- ► Effective time management skills and ability to work on multiple projects simultaneously.

Visit UA's staff employment website at jobs.ua.edu for information and to apply.

The University of Alabama is an equal-opportunity employer (EO), including an EOE of protected vets and individuals with disabilities.

University of Maine

School of Computing and Information Science

Tenure-Track Assistant Professor of Computer Science

The University of Maine School of Computing and Information Science seeks applicants for a tenure-track, academic-year Assistant Professor position, with an anticipated start date of September 1, 2019. Our primary target focus is the broad area of data science (machine learning, data mining, data management, information retrieval, natural language processing, computer vision, data visualization, etc.), but we will consider exceptionally qualified candidates in all areas.

Knowledge, skills, and qualifications. We are particularly interested in candidates who complement the School's existing research strengths in Artificial Intelligence (AI), Data Management, Distributed Systems, Human-Computer Interaction (HCI), Cybersecurity/Privacy, Data Visualization, and Spatial Informatics with potential for collaborations within the School. The ability to contribute to campus-wide Signature and Emerging areas of excellence in Data Science and STEM Education Research also would be viewed favorably. Candidates should have a record of research excellence as demonstrated by relevant and recent publications in top peer-reviewed conferences or journals, presentations at significant conferences, and evidence of potential for success in attracting external funding. Successful teaching experience and documented participation in activities promoting inclusive excellence are also highly desired. A Ph.D. in computer science or a closely related discipline is required by date of hire, as is documented recent research.

Essential duties and responsibilities. The successful candidate will be expected to establish a productive research program in their field of expertise; to be an engaging teacher, adviser, and mentor at both the undergraduate and graduate levels; and to make a strong commitment to the development of innovative computing curricula. The typical teaching load is three courses per year, including both undergraduate and graduate courses. Service to the School, College, University, and profession is expected.

The School (umaine.edu/scis) is an interdisciplinary unit encompassing Computer Science, Spatial Informatics, and New Media and offering degrees in Computer Science (BS, BA, MS, Ph.D.), Spatial Information Science and Engineering (MS, Ph.D.), Information Systems (MS), and New

Media (BA). This disciplinary breadth results in abundant opportunities for easy and exciting collaboration. The School has 20 full-time faculty members and enrolls approximately 300 undergraduate and 45 graduate students. In addition to the tenure-track assistant professor position, our growing School is also currently recruiting a full-time lecturer. The University of Maine is the flagship campus of the University of Maine System and is the principal graduate institution in the state. It is the state's land and sea grant university, enrolling over 11,000 students. The University of Maine is home to the Rising Tide Center, launched with support from NSF's ADVANCE program and continuing its mission to advance gender equity. Numerous cultural activities, excellent public schools in neighborhoods where children can walk to school, high-quality medical care, little traffic, and a reasonable cost of living make the greater Bangor area a wonderful place to live. The University is located just 60 miles from the scenic Bar Harbor area and Acadia National Park and two hours from Portland.

Increasing the diversity of the computing profession is one of our strategic priorities. Women and those from groups traditionally underrepresented in computing fields are particularly encouraged to apply.

To apply, visit bit.ly/UMaineCSTTPosition (or check umaine.hiretouch.com). A complete application includes a letter of application, CV, teaching statement, research statement, and list of references. The letter of application should directly address the applicant's qualifications as described above. Please also provide contact information for at least three references who can address the applicant's research and teaching qualifications and accomplishments. Letters of reference may be requested at a later stage. Questions may be directed to Dr. Silvia Nittel, Chair, CS Faculty Search Committee, University of Maine, 5711 Boardman Hall, Orono, ME 04469-5711 or silvia.nittel@maine.edu. Incomplete application materials cannot be considered. Review of applications will begin 1/31/2019 and continue until the position is filled.

The University of Maine, an EO/AA employer, seeks to employ outstanding people who contribute to the rich cultural diversity expected in a university setting. All qualified individuals are encouraged to apply.

Washington State University Vancouver

Tenure-Track Position at the Assistant Professor Level

Computer Science Faculty - Washington State University Vancouver invites applications for a full-time tenure-track position at the assistant professor level beginning 8/16/2019. Candidates from all areas of computer science, including theory, will be considered with preference given to expertise in **computer networks**, **wireless networks** or **sensor networks**.

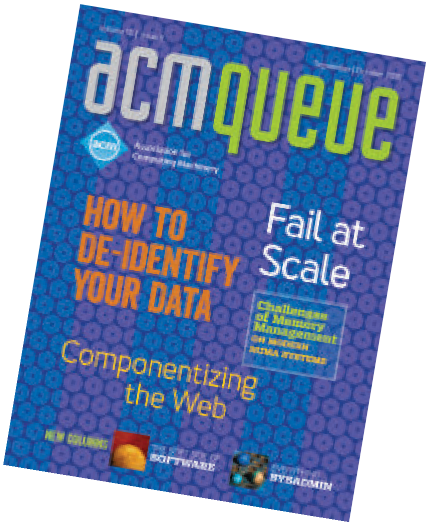
Required qualifications: Ph.D. in Computer Science or Software Engineering by the employment start date and demonstrated ability to (1) develop a funded research program, (2) establish industrial collaborations, (3) teach undergraduate/graduate courses, and (4) have published promising scholarly work in the field and (5) contribute to our campus diversity goals (e.g. incorporate issues of diversity into mentoring, curriculum, service or research).

Duties include: (1) Teach undergraduate and graduate courses including networks; (2) Conduct research in at least one of the expertise areas listed above; (3) Secure external funding for research; and (4) Participate in service to the department and university through committee work, recruitment, and interaction with industry.

WSU Vancouver serves about 3,400 graduate and undergraduate students and is **fifteen miles north of Portland, Oregon**. The rapidly growing School of Engineering and Computer Science (ENCS) **equally values both research and teaching**. WSU is Washington's land grant university with faculty and programs on five campuses. For more information: http://ecs.vancouver.wsu.edu. WSU Vancouver is committed to building a culturally diverse educational environment.

Application: Please visit www.wsujobs.com and search postings by location. Applications must include: (1) cover letter with a clear description of experience relevant to each of the required and preferred qualifications; (2) vita including a list of at least three references; (3) a statement (two-page total) of how candidate's research will expand/complement the current research in ENCS and a list of the existing ENCS courses the candidate can teach and any new courses the candidate proposes to develop; and (4) a statement on equity and diversity (guidelines found at https://admin.vancouver.wsu.edu/sites/admin.vancouver.wsu.edu/files/Diversity%20Statement%20Guidelines.pdf). Application deadline is **April 7, 2019**.

WASHINGTON STATE UNIVERSITY IS AN EQUAL OPPORTUNITY/AFFIRMATIVE ACTION EDUCATOR AND EMPLOYER. Members of ethnic minorities, women, special disabled veterans, veterans of the Vietnam-era, recently separated veterans, and other protected veterans, persons of disability and/or persons age 40 and over are encouraged to apply. WSU employs only U.S. citizens and lawfully authorized non-U.S. citizens.


acmqueue
HOW TO DE-IDENTIFY YOUR DATA
Fail at Scale
Challenges of Memory Management on Modern NUMA Systems
Componentizing the Web
NEW COLUMNS
SOFTWARE
SYSADMIN

[CONTINUED FROM P. 120]

tant professor, along with many other people in the field. During that era, there was a huge effort to design machine learning models that could recognize objects. We also had to find sensible ways to benchmark their performance. And there were some very good datasets, but in general they were relatively small, with only one or two dozen different objects.

When datasets are small, it limits the type of models that can be built, because there's no way to train algorithms to recognize the variability even of a single object like "cat."

People were making progress in that era, but the field was a little bit stuck, because the algorithms were unsatisfying. So around 2006, my students and I started to think about a different way of approaching the object recognition problem. We were thinking that instead of designing models that overfit on a small dataset, we would think about very large-scale data, like millions and millions of objects, and that would drive machine learning models in a whole different direction.

So you started working on ImageNet, which seemed crazy at the time.

Our goal was to map out all the nouns in the English language, then collect hundreds of thousands of pictures to depict the variability of each object, like an apple or a German Shepherd. We ended up downloading and sifting through at least a billion pictures or more, and we eventually put together ImageNet though crowdsourcing. That dataset was 15 million images and 22,000 object categories.

In your research at Stanford's Vision and Learning Lab, you work closely not just with technologists, but also with neuroscientists. Can you tell me a bit about how that collaboration works?

Fundamentally, AI is a technical field. Its ultimate goal is to enable machine intelligence. But because human intelligence is so closely related to this field, it helps to have a background and collaborators in neuroscience and cognitive science. Take today's deep learning revolution. The algorithms we use today in neural networks were inspired by classic studies of neuroscience back in the 50s and 60s, when scientists found neurons are layered together and send information in a hierarchical way. In the meantime, cognitive science has always been an essential part of guiding AI's quest for different kind of tasks. Many computer scientists were inspired to work on object recognition, for example, because of the work cognitive scientists had done.

> "Our goal was to map out all the nouns in the English language, then collect ... pictures to depict the variability of each object, like an apple or a German Shepherd."

One of your current interdisciplinary collaborations is a neural network that implements curiosity-driven learning.

Human babies learn by exploring the world. We are trying to create algorithms that bear those kinds of features—where computers go where they go out of curiosity rather than being trained on traditional tasks like labeled images.

You have spoken before about the need to think about AI from a humanistic and not just a technical perspective, and you just helped launch Stanford's Human-Centered AI Initiative (HAI). Can you talk about your goals?

We want to create an institute that works on technologies to enhance human capabilities. In the case of robotics, machines can do things humans cannot. Machines can go to dangerous places. They can dive deeper in water and dismantle explosive devices. Machines also have the kind of precision and strength humans do not. But humans have a lot more stability and understanding, and we have an easier time collaborating with one another. There are a lot of potential scenarios we can imagine in the future where machines assist and augment humans' work, rather than replacing it.

You've also been vocal about the need to include a more diverse set of voices in computer science and AI research.

If we believe machine values represent human values, we need to believe we have fully represented humanity as we develop and deploy our technology. So it's important to encourage students of diverse backgrounds to participate in the field. It's also important, at this moment, to recognize the social impact of technology is rising. The stakes are higher than ever, and we also need to invite future business leaders, policymakers, humanists, social scientists of diverse backgrounds to be technologically literate, to interact with the tech world, and to bring that diverse thinking into the process.

Can you tell me about Stanford's new AI4All program for high school students, which grew out of the earlier Stanford Artificial Intelligence Laboratory's Outreach Summer Program (SAILORS)?

AI4All aims to increase diversity in the field of artificial intelligence by targeting students from a range of financial and cultural backgrounds. It's a community we feel very proud of and are very proud to support. One of our earliest SAILORS alumna, a high school student named Amy Jin, continued working in my lab on videos for surgical training. Then, while still in high school, she authored a research paper with my team that was selected by the 2017 Machine Learning for Health Workshop's Neural Information Processing Systems (NIPS) conference, one of the best-respected events in the field. What's more, out of 150 papers, she won the award for best paper. We also have students who started robotics labs at their schools and hold girl-centered hackathons. Many of them are focusing on applications that put AI to good social use, from optimizing ambulance deployment to cancer research and cyberbullying.

Leah Hoffmann is a technology writer based in Piermont, NY, USA.

DOI:10.1145/3303853

Leah Hoffmann

Q&A
Guiding Computers, Robots to See and Think

Fei-Fei Li, co-director of Stanford University's Human-Centered AI Institute, wants to create algorithms that can learn the way human babies do.

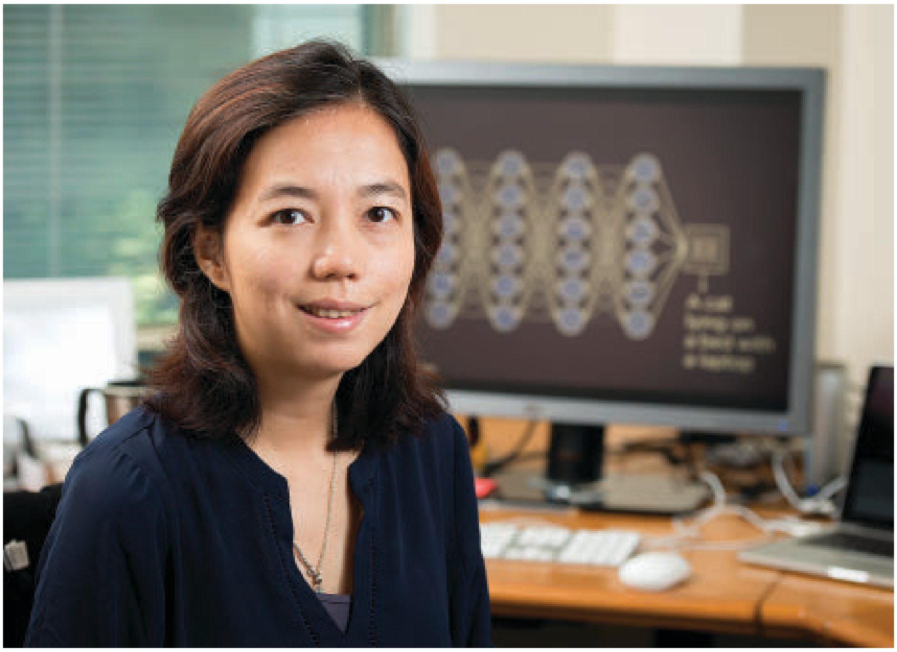


PHOTO BY LINDA A. CICERO/STANFORD NEWS SERVICE

THOUGH STANFORD UNIVERSITY professor Fei-Fei Li began her career during the most recent artificial intelligence (AI) winter, she's responsible for one of the insights that helped precipitate its thaw. By creating Image-Net, a hierarchically organized image database with more than 15 million images, she demonstrated the importance of rich datasets in developing algorithms—and launched the competition that eventually brought widespread attention to Geoffrey Hinton, Ilya Sutskever, and Alex Krizhevsky's work on deep convolutional neural networks. Today Li, who was recently named an ACM Fellow, directs the Stanford Artificial Intelligence Lab and the Stanford Vision and Learning Lab, where she works to build smart algorithms that enable computers and robots to see and think. Here, she talks about computer vision, neuroscience, and bringing more diversity to the field.

Your bachelor's degree is in physics and your Ph.D. is in electrical engineering. What drew you to computer vision and artificial intelligence (AI)?

When I was an undergrad at Princeton, I had a lot of academic freedom. By sophomore year, I was already fascinated by the writings of physicists from the early 20th century—people like Schrödinger and Einstein who, in the later part of their careers, all had a lot of curiosity about life and intelligence. Then I did a couple of research projects related to neuroscience and modeling; I was hooked. I decided to pursue a Ph.D. in a combination of cognitive neuroscience and computer vision—we didn't call it AI at that point.

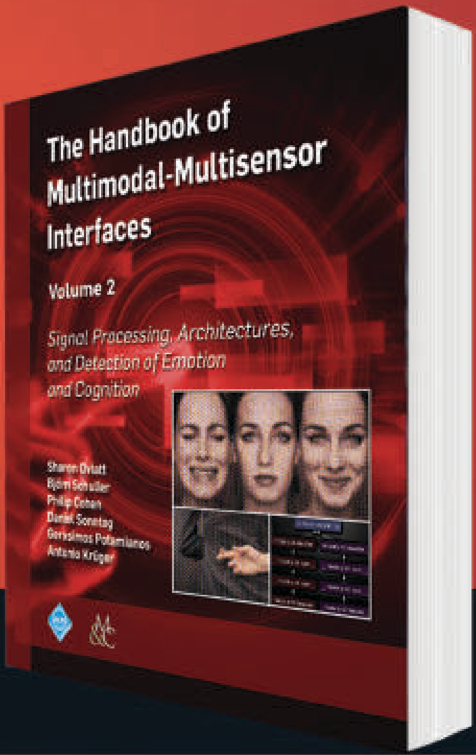
This was during one of the so-called AI winters, when interest and investment cooled as people realized technologies had failed to live up to their hype.

While I was studying for my Ph.D., it was a very interesting time. Machine learning became a very important tool in computer vision, so I was in the generation of students who got a lot of exposure and training in that subject.

That training helped crystallize an insight that proved pivotal to the field of AI, namely that creating better datasets would help computers make better decisions. This prompted you to build ImageNet, a hierarchically organized image database in which each node of the hierarchy is depicted by hundreds and thousands of images.

In the field of AI, there are a few important problems that everyone works on; we call them 'holy grail' problems. One of them is understanding objects, which is a building block of visual intelligence. Humans are superbly good at recognizing tens of thousands and even millions of objects, and we do it effortlessly on a daily basis. So I was working on this problem when I was a Ph.D. student and in my early years as an assis-

[CONTINUED ON P. 119]


The Handbook of Multimodal-Multisensor Interfaces
Volume 2
Signal Processing, Architectures, and Detection of Emotion and Cognition
Sharon Oviatt
Björn Schuller
Philip Cohen
Daniel Sonntag
Gerasimos Potamianos
Antonio Krüger