When We Search the Web

What Does the Relationship between Algorithms and Information Have To Do with Our Everyday Computer Use?

ROBERT KLEINBERG, COMPUTER SCIENCE

Network Mapping Camp

My research is on understanding the relationship between algorithms and information. Algorithms are the computational processes by which we process information and use it to make decisions. To illustrate the type of research I became excited about, I should first tell another story. My major at Cornell was math. I went to graduate school at MIT to study pure math. My PhD is in math. After my second year at MIT, I decided to take a summer job. The year was 1999. Boston was full of tech start-ups, many of them spin-offs from MIT. In particular there was a spin-off from my home department called Akamai Technologies, which has become one of the big players in internet services. It was founded by a MIT professor who became my thesis adviser.

The company provides a service that makes other websites more scalable, fast, and secure. They accomplish this with a massive distributed network of tens of thousands of servers around the world. (We have some of them on the Cornell campus, for example.) If you are an Akamai customer, the content on your website gets spread out in these thousands of locations so that when people download it, it comes to them from a computer that is much closer to them. If an attacker tries to take down your site, instead of taking down one computer, they have to take down 10,000. If you have a power failure that takes down your site, a power failure in one place cannot take out a site that is distributed over a thousand locations.

I worked in a group called the network mapping group. If you have servers in thousands of locations around the world, and you are supposed to direct every user to the one that is best suited for serving their request, this is a gigantic optimization problem to solve—millions and millions of users and thousands of locations. My group, the mapping group, was in charge of supplying the input to that optimization problem. We wanted to be able to tell the decision-making apparatus the information it needed to know in order to make the best possible decisions.

Traditionally, one would take for granted that this information is easy to obtain and that the hard part is processing it and figuring out where each person should be routed. In reality, it is an incredibly tough engineering challenge to measure the relevant information for making this decision. This requires measuring the network conditions between each of our servers and every client. We would be performing billions of measurements every second, which is not technologically feasible.

Jason Kosky/CU

What information do we actually need in order to make the best decisions, and what is a smart and scalable strategy for acquiring this information? This is what I mean when I talk about the relationship between algorithms and information. We live in an age where most of the algorithms that ordinary people care about in order to make optimal decisions require a greater amount of information than is technologically feasible for them to obtain.

[As a search engine], if I notice people always jump over the number one spot and pick the number two spot, eventually I should learn to swap them.

Google It

Think about using Google. You enter a query into Google, and the optimal thing for Google to do is to know the tastes of every single user in the world and present the search results that are most relevant to your query. For every user, query, and document, Google would need to be able to measure relevance. It is impossible. So what is the minimum set of information needed in order to make the right decision about presenting search results, and what is a smart strategy for acquiring that information? Increasingly these two questions are inseparable—the question of how to make the right decisions given perfect information, and how to acquire information. This has become a closed loop. The information acquisition process has to be informed by the decision-making process in a feedback loop, rather than the traditional method of running experiments to completion, getting a set of data, and then doing the decision making on that data.

The reasons why information might be unavailable to decision makers are many. One is that the scale of the problem makes it technologically impossible to acquire all the relevant information. The other is that the information might belong to self-interested parties who do not want to reveal it to you unless it is in their interest to do so, or they might want to give you misinformation if they think they can benefit from doing that. The game theoretic aspect of this problem is to create a system that encourages the different players in control of the information to participate and reveal accurate, non-misleading information.

What Price Should I Charge?

One of the first things I worked on in graduate school was pricing problems. Let's say that I have written a new piece of software, and I am going to sell it online. Because it is software, it is a digital good, and it costs me zero to create extra copies of the software once I have written it. However, I do not want to give it away for free; I want to charge a price. I have no idea what the demand is for my product. How do I figure out what price to charge? A traditional solution would be to hire a market research firm and conduct a survey to gauge the demand in the market, set a price, and post that price on my website, and a certain number of people would buy the product from me.

A more automated solution would be to run my own price experiments on the site. Every time a new person comes I would present a new price, which might be different from any price I have charged before. They decide to click or not to click, and that gives me one more data point in my search path to finding the optimum price point. Suppose you want to do this kind of price experimentation. What is the optimum strategy for setting the sequence of prices so that the opportunity cost of learning the optimum price is minimized? How much better or worse off are you by doing this type of price experimentation, rather than paying the outside marketing research firm to give you the data? The answer to these questions is surprisingly robust.

What is the value of knowing the information on day one? One way I could put the question to you is this: If there's a person right here with a briefcase who knows exactly what price maximizes my profits, and that price is written down inside the

Fascinating!

- What information do we actually need in order to make the best decisions, and what is a smart and scalable strategy for acquiring this information? This is what I mean when I talk about the relationship between algorithms and information.
- You enter a query into Google, and the optimal thing for Google to do is to know the tastes of every single user in the world.
- An application that has gained my interest recently is ranking web search results.
- Say I am a search engine.... If I have a query with multiple meanings, where "jaguar" could refer to an animal, a car, or a football team from Jacksonville, how do I learn that it has multiple meanings?
- How do we design learning algorithms that work well in tandem with each other, assuming each device uses the algorithm?

Kleinberg in Duffield Hall with (r.) Hu Fu and (l.) Bruno Abrahao

briefcase, how much should I be willing to pay him on day one to open up the briefcase, given that my outside option is to run an algorithm and experiment to find the optimum price point? The answer turns out to be roughly the square root of n, where n is the number of users who will ever visit my site. It is remarkably robust because it remains valid under almost any reasonable set of assumptions about the user population. The answer might be 10 times the square root of n; it might be 50 times the square root of n; but it always scales to the market size according to the square root of n.

My adviser and I made this discovery on the way to producing my doctoral thesis. The same techniques I employed in answering that question have fed into other computer science problems in which my colleagues and I have an interest.

Searching the Web

I still work on the same kind of algorithmic experimentation problems, but an application that has gained my interest recently is ranking web search results. Say, for instance, I am a search engine. I get queries, then I present rankings to people, and they click somewhere within the ranking. I want to data mine the clicks I have experienced over time to improve my ranking. Take an obvious example: if I notice people always jump over the number one spot and pick the number two spot, eventually I should learn to swap them.

What is the optimum sequence of ranking experiments to rapidly rerank the search results and present the most relevant documents early? How do things like diversity of user tastes influence this search process? If I have a query with multiple meanings, where "jaguar" could refer to an animal, a car, or a football team from Jacksonville, how do I learn that it has multiple meanings? How do I learn to present at least one example of each high in the rankings so that people with different information needs are all satisfied by the search results? How do I choose the optimal sequence of experiments in order to learn this information? These would be my problems to solve.

No Clobbering Allowed

Another project relates to multiple learners interacting with each other at the same time. Let's think, for example, about the wireless network in this building (Upson Hall). We have multiple wireless networks with different channels, and given the current utilization of the network, we get a clean connection regardless of the channel we choose. In the future, however, when we have hundreds of wireless devices operating simultaneously, we could have congested and uncongested channels.

Imagine that every device runs an experimentation algorithm where it hops around from one channel to another trying to find one where it can get a clean signal under the current conditions. This is acceptable only if one device is doing it, and the others are staying put. But if all devices are doing it at the same time, bad feedback effects emerge where they respond in tandem to a problem on one channel and clobber another one. How do we design learning algorithms that work well in tandem with each other, assuming each device uses the algorithm? If they all do it together, they will not get into a bad feedback effect that clobbers each other's efforts to find the optimal configuration. I am working on this problem with a colleague here in computer science.

It's Second Nature @ Cornell

When I first arrived here as a faculty member, it was striking how people—not only in computer science but also in electrical engineering, operations research, and others—would come to hear my talks. They brought great ideas about how my topic tied in with their work and how we could work together. This happens so often at Cornell that we treat it as second nature. I have been at other places that did not have this culture. People push their own research and excel in it, but they do not have the same outward-looking attitude where one could come into another person's talk, listen for an hour, and find a connection between what they do and what that person does. These unexpected interactions have been such an enriching factor in my own life.

This is woven into a culture like Cornell's. Senior faculty who have been here for 25—and in some cases 40—years still listen to junior faculty and find ways to tie in. To keep your eyes wide open like this all the time is a lot of work, but it benefits everybody and makes this a more fun place to work. What motivates me to walk up the hill every morning is knowing that I am going to have these interactions.

From Abstract Math Problems to Yahoo Users to Higher Priorities

As early as I can remember, I have loved math. This love of math has been passed down through my family. When we went on long vacations together in the car, part of the time would be spent talking about math. Math puzzles always floated around the dinner table. Very early on, I appreciated how much fun solving puzzles about numbers and geometry could be, and by my teens, I knew that math could be a very satisfying career. An important turning point came when I stepped outside of the sheltered abstract world of math to experience life in a start-up—a polar opposite of the math research process.

One day I was sitting at my desk at MIT working on abstract problems that only five or six other people in the world would understand or even care about. Two days later, I had an office at Akamai with a team of extremely energized people who knew that the software they were writing today

About Kleinberg

Years as Cornell faculty
2

Came to Cornell from
University of California–Berkeley

Favorite spot on campus
Olin Library terrace and Duffield atrium

Cornell's research distinction
A pervasive team spirit

Cornell's trademark
Collaboration

I am also
A classical pianist

would be affecting every Yahoo user in the world next week. It is hard to overstate how radical and inspiring it felt to be part of that and working on something that was so rapidly and directly impacting millions of people. It was intoxicating. I knew that I wanted to have more of that.

The next turning point came with my decision to move back to academics instead of staying in the company. Working in academia prioritizes a different type of question, and it is one that suits my personality much better. I have worked on difficult and fun math problems in industry and academia. Here is the difference: in industry, the importance of a question and its solution is judged on its usefulness. Will it make this thing work better? Will it make us more money? In academia, the importance of the question and its solution is judged on long-term impact. Will this question teach us something novel we do not understand yet? In the end, these solutions are profoundly useful, so it is not as if we sacrifice usefulness. We take a view, which is slanted toward prioritizing questions for conceptual elegance and advancing human knowledge rather than immediate operational usefulness. This suits me better. I knew that my path toward self-fulfillment would lie in academia, not industry.

A bonus is that I discovered how much I love teaching. It is so much fun to see how smart and quick Cornell students are. If I make a mistake in my lecture, the students not only catch my mistake, but one of them will have figured out how to fix it by the time I get back to it.

Where the Action Is

Cornell's excitement is all about the excellence of its people and the team spirit that pervades my department and the whole university. Academia is so often about individual excellence, so it is unusual and satisfying to be part of a department that feels like a team where every person contributes to the larger goal of building an excellent computer science department.

Cornell is one of the places that the entire world looks to for leadership on defining the foundation of a new kind of computer science. When I was in graduate school at MIT, so much of the research I read and heard about came out of Cornell's CS department, which has been on the leading edge of trends that propagated to other top institutions. A concrete example is game theory, which is becoming an increasingly important aspect of computer science.

Computation now deals with interactions among multiple users online and among multiple software processes online, and they have conflicting interests. We cannot assume that they will cooperate with each other. Ten or 20 years ago, the science was about how to get a single process to operate as efficiently as possible on its assigned task. Today we are focused on creating protocols that promote positive interaction among processes whose individual self-interests may be in conflict. My career and research trajectory led me naturally into those questions. Many of the papers I read came out of Cornell, which remains on the leading edge of the trend.

> As early as I can remember, I have loved math. This love of math has been passed down through my family.

Favorite Spots on Campus

From the Olin Library terrace, I can look out over the Arts Quad. This is a very special place to me. Another place I especially love on campus is the Duffield atrium. Having this space where we can congregate and relax has added so much to the engineering college. The computer science department takes such advantage of the space. Our department typically has more than 10 people congregated around a couple of tables in the atrium every day for lunch.

Living in Ithaca

The Ithaca Way. Ithaca is one of the most beautiful places I have ever been, and the beauty is integrated into the Cornell campus. My walk to work each morning is though Cascadilla Gorge. We are right in the middle of a vibrant city and university, but walking through the gorge, I feel like I am in a different place and time. Prior to Ithaca, I lived in Berkeley. I was a postdoc in computer science working on game theory for slightly more than a year. Even though Ithaca and Berkeley have a lot in common, I identify more with the Ithaca way. I love the kind of people who choose to live here, and it is not just the university faculty and their families, but also the people I meet downtown and throughout the community.

A Wish List for Ithaca. My wish list includes a good opera company, an Ethiopian restaurant, and a larger number of airlines to make it easier to travel in and out of Ithaca—more convenient for us to travel and others to visit us.

Hobbies. I enjoy playing piano, running, and tennis. I play classical piano, and my favorite composers are Beethoven and Schumann. As an undergraduate, I studied piano seriously, spending almost as much time on piano as math. Since then, I have continued it as a hobby. I have a piano at home, and when I need to get away from my work and relax, it is one of my favorite activities.

The Last Word

Timeless Cornell

I feel that I am part of something bigger than the here and now, and bigger than the individual efforts of each person, at Cornell. The people—students, faculty, and staff—and the energy of team effort characterize the university. As I walk around the campus in all of its space, there is a feeling of timelessness.

For more information:
E-mail: rdk2@cornell.edu
Website: www.cs.cornell.edu/~rdk