

Hector Garcia-Molina — Interests 2011

I am a member of the Stanford [InfoLab](#), together with my colleagues [Jennifer Widom](#) and [Jure Leskovec](#) (plus emeritus members [Jeff Ullman](#) and [Gio Wiederhold](#)). Our InfoLab focuses on *information management*: all types of information, from structured data in traditional databases, to unstructured, media-rich information at sites like Twitter and Facebook. And we focus on all types of management, from collection and extraction of the data, to its storage, to its analysis.

On this Web page I give some examples of my current interests. Note that even though not reflected directly in these examples, I continue to be interested in traditional problems dealing with database management systems, distributed data management, digital libraries, and peer-to-peer systems. Also note that my interests are evolving all the time, often driven by what new problems students or industry colleagues introduce me to, so for the latest, stop by and chat!

For each example, I give a brief summary plus one or two references to recent papers. Click on the link below or just scroll down.

- [Web Analytics](#): Discovering trends and patterns in Web and social network data.
- [Crowdsourcing](#): Using humans as a source of information.
- [Electronic Commerce](#): Advertising, selling and trading goods and information.
- [Entity Resolution](#): Matching information fragments on the same entity (“connecting the dots”).
- [Recommendations](#): Discovering interesting and unexpected information.

Web Analytics

The Web contains a wealth of information, and there are many challenges in getting the data, interpreting it, organizing it, and learning from it. For instance, one type of information is the tags that people associate with their photos, web pages and other resources. (For example, a photo may be tagged with the words “San Francisco” or an Amazon book may be tagged with “romantic novel”.) How useful and accurate are such tags? Do they really help people find things they want? In the paper [“Tagging Human Knowledge”](#) we compared the tags that people attached to books at several social cataloguing sites, to tags added by professional librarians. Do you think unpaid volunteers at the social sites do a better job than the librarians? A more comprehensive job? To find out, read the paper!

There are many different types of data that we are analyzing in the InfoLab, but often the challenge is in actually getting the data. One of the main problems is that sources may be unwilling to give you a full data set (or not able to due to the high volume), and they instead give you a sample of the data. This was the case with data we were getting from Twitter: it only represented about 10% of the tweets made over a couple of months (and we are not even sure what the actual fraction was). So, how does the fact you are analyzing sampled data affect your results? For instance, say you are trying to predict how fast a bit of news propagates across the Twitter social graph. If your analysis says 1 hour but you used a 10% sample, is the correct answer 10 hours? Or a tenth of an hour? Or the same one hour? And is there any way of knowing, just by looking at the sample, what was the actual sampling rate? To get some answers, see [“Correcting for Missing Data in Information Cascades”](#).

CrowdSourcing

People are another information resource, and in the InfoLab we are exploring how we can combine a “crowd” (lots of people) source with traditional sources. Sites such as Mechanical Turk make it possible to ask people (for a small fee) for information, or to perform tasks that are hard for computers to do. For example, say we want to build a table (relation) for restaurants. The attributes (columns) for such a table include the name of the restaurant, the address, the city, the rating and the cuisine. Some of the attributes (e.g., name and address) could come from a traditional stored database, while the others may come from the crowd, i.e., may be the result of asking people for their opinions. We are currently building a database system, DECO, that can support such a table. The end user will be unaware that some of the attributes are crowd sourced. He will be able to pose queries, using standard SQL (well, almost standard). For instance, he may ask for “four-star Chinese restaurants in Palo Alto”. If the system does not have the ratings or the cuisine for some Palo Alto restaurants, it will on the fly issue requests to humans (e.g., via Mechanical Turk) to get the missing data. Note that in many cases multiple requests will need to be issued to compute the average rating of a restaurant, or the consensus cuisine. If there are not enough Palo Alto restaurants in the database, DECO may even asks humans for the names of additional restaurants. One of the important challenges here is in finding a “good” plan for answering a query, as we may wish to minimize the number of crowd requests made, or the amount of time taken to get some minimal number of answer restaurants, and with a given level of “quality” (e.g., number of votes used to compute a rating). We are currently writing a paper that describes DECO, and I will post it here as soon as it is ready.

In addition to the database aspects of crowdsourcing, we are also exploring some of the algorithmic challenges. To illustrate, say we have a set of 1000 photos and we are trying to find the best one, for some definition of best. We can issue requests to humans to compare 2 photos and tell us which one they like best. (There are many other choices for the crowd interface: for instance, we could show humans 5 photos and ask them to order them by their preference.) What is the best strategy for finding the best overall photo? Keep in mind that humans can make mistakes, so we may need to perform the same comparison multiple times. As a first step, should we compare all pairs of photos? Or only some pairs, and based on the results, compare additional pairs? Should we try to figure out which humans are more reliable, so we can weight their votes more? We are addressing these types of questions for several tasks (e.g., sorting, finding the top-k elements, classification, etc). Again, a report is being written and will be posted soon.

Electronic Commerce

Commerce is one of the principal uses of the Web, so we have also been exploring how people buy and sell products or information electronically. Here are two examples of our work. You are probably all familiar with advertising by search engines: You type in the query “digital camera” and in addition to your results, you get “sponsored links” (at the top of the results or on the right) for different stores and camera manufacturers. We call this *input bidding* since the advertisers bid on your input (the query). The more an advertiser bids, the more likely he is to win the auction and have his ad shown on the results page. Together with researchers at Yahoo, we have been exploring a complementary approach,

output bidding. Here the advertiser bids on URLs that appear on the results page. For example, an advertiser may want his ad to appear whenever the search result includes the sites www.imdb.com and en.wikipedia.org, instead of bidding on keywords that lead to these sites, e.g., movie titles, actor names, and so on. We claim that output bidding is more compact than input bidding, e.g., fewer URLs are needed to represent the interests of an advertiser. But if you do not believe me, read this paper [“Output URL Bidding”](#).

Another popular type of advertising is display advertising. Here “banner ads” appear at the top or side of the page, whenever a page is visited. The choice of ad to display depends on the user visiting the page. For instance, an advertiser may request that a particular ad be displayed whenever a young male living in California visits a page at the New York Times. Advertisers are always interested in knowing if their campaign is effective, that is, are people that see the ad more likely to buy their product or visit their Web site. Again with Yahoo, we have been evaluating the effectiveness (lift) of campaigns, by seeing if people who see an ad for X are then more likely to ask queries related to X. Furthermore, we are exploring the influence of social networks: If person A sees the ad for X, and A sends an email to B, is then B more likely to ask queries related to X? Is the social influence greater if A has more (or fewer) friends? How long does the influence last? Hours after the ad is displayed? Or days? The answers to questions like these are very important to Web companies and their advertisers, as they can determine how much can be charged for advertisements (money that can then be used to provide free services to others ☺). We try to answer some of these questions in [“Display Advertising Impact: Search Lift and Social Influence”](#).

Entity Resolution

Entity Resolution (ER) is the process of matching information records (typically from different sources) that may refer to the same real-world entity. Matching records are often merged into “composite” records that reflect the aggregate information known about the entity. A current direction we are pursuing is the relationship between ER and information privacy. As more of our sensitive data gets exposed to a variety of merchants, health care providers, employers, social sites and so on, there is a higher chance that an adversary can “connect the dots” and piece together our information, leading to even more loss of privacy. For instance, suppose that Alice has a social networking profile with her name and photo and a web homepage containing her name and address. An adversary Eve may be able to link the profile and homepage to connect the photo and address of Alice and thus glean more personal information. The better Eve is at linking the information, the more vulnerable is Alice’s privacy. Thus in order to gain DP, one must try to prevent important bits of information being resolved by ER. In the paper [“Managing Information Leakage”](#) we study this problem and list a number of interesting open problems in this area. Take a look!

Recommendations

Several years ago a few InfoLab students built a social networking site for Stanford students, CourseRank, which was quite successful. ([CourseRank](#) is now commercial, and runs at over 300 universities.) We used CourseRank as a testbed for our research, focusing on the course

recommendations aspect. See for example, "[Recsplorer: Recommendation Algorithms based on Precedence Mining](#)". More recently, we were asked by the Communications of the ACM to write an essay on current trends and future challenges for recommendation systems, given our CourseRank experience. In the paper "[Information Seeking: Convergence of Search, Recommendations and Advertising](#)", we argue that recommendation mechanisms are "converging" with search and advertising mechanisms, and that we should really design all "information seeking" strategies in a unified way. Not everyone agrees with our thesis; some say a unified mechanism for search, recommendations and advertising would be too inefficient. What do you think??