Deconstructing Recommender Systems

How Amazon and Netflix predict your preferences and prod you to purchase

By Joseph A. Konstan, John Riedl
Posted 24 Sep 2012 | 14:30 GMT

Photo-illustration: Christina Beard

One morning in April, we each directed our browsers to Amazon.com (http://amazon.com)'s website. Not only did the site greet us by name, the home page opened with a host of suggested purchases. It directed Joe to Barry Greenstein's *Ace on the River: An Advanced Poker Guide,* Jonah Lehrer's *Imagine: How Creativity Works*, and Michael Lewis's *Boomerang: Travels in the New Third World*. For John it selected Dave Barry's *Only Travel Guide You'll Ever Need*, the spy novel *Mission to Paris*, by Alan Furst, and the banking exposé *The Big Short: Inside the Doomsday Machine*, also by Michael Lewis.

By now, online shoppers are accustomed to getting these personalized suggestions. Netflix (http://Neflix.com) suggests videos to watch. TiVo records programs on its own, just in case we're interested. And Pandora (http://Pandora.com) builds personalized music streams by predicting what we'll want to listen to.

All of these suggestions come from recommender systems. Driven by computer algorithms, recommenders help consumers by selecting products they will probably like and might buy based on their browsing, searches, purchases, and preferences. Designed to help retailers boost sales, recommenders are a huge and growing business. Meanwhile, the field of recommender system development has grown from a couple of dozen researchers in the mid-1990s to hundreds of researchers today—working for universities, the large online retailers, and dozens of other companies whose sole focus is on these types of systems.

Over the years, recommenders have evolved considerably. They started as relatively crude and often inaccurate predictors of behavior. But the systems improved quickly as more and different types of data about website users became available and they were able to apply innovative algorithms to that data. Today, recommenders are extremely sophisticated and specialized systems that often seem to know you better than you know yourself. And they're expanding beyond retail sites. Universities use them to steer students to courses. Cellphone companies rely on them to predict which users are in danger of switching to another provider. And conference organizers have tested them for assigning papers to peer reviewers.

The two of us have been building and studying recommender systems since their early days, initially as academic researchers working on the GroupLens Project (http://www.grouplens.org/). Begun in 1992, GroupLens sorted through messages in Usenet discussion forums and pointed users to threads they might be interested in but had not yet discovered on their own. Several years later, we founded Net

Perceptions (http://www.networkworld.com/newsletters/web/0503web2.html), the leading recommender company during the first Internet boom. Our experience, therefore, gives us a lot of insight into what's going on behind the scenes at Amazon and other online retailers, even though those companies seldom speak publicly about exactly how their recommendations work. (In this article, our analysis is based on educated observation and deduction, not on any inside information.) Here's what we know.

Have you ever wondered what you look like to Amazon? Here is the cold, hard truth: You are a very long row of numbers in a very, very large table. This row describes everything you've looked at, everything you've clicked on, and everything you've purchased on the site; the rest of the table represents the millions of other Amazon shoppers. Your row changes every time you enter the site, and it changes again with every action you take while you're there. That information in turn affects what you see on each page you visit and what e-mail and special offers you receive from the company.

Over the years, the developers of recommender systems have tried a variety of approaches to gather and parse all that data. These days, they've mostly settled on what is called the personalized collaborative recommender. That type of recommender is at the heart of Amazon, Netflix, Facebook's friend suggestions, and Last.fm (http://www.last.fm/), a popular music website based in the United Kingdom. They're "personalized" because they track each user's behavior—pages viewed, purchases, and ratings—to come up with recommendations; they aren't bringing up canned sets of suggestions. And they're "collaborative" because they treat two items as being related based on the fact that lots of other customers have purchased or stated a preference for those items, rather than by analyzing sets of product features or keywords.

Personalized collaborative recommenders, in some form or another, have been around since at least 1992. In addition to the GroupLens project, another early recommender was MIT's Ringo (http://www.sigchi.org/chi95/proceedings/papers/us_bdy.htm), which took lists of albums from users and suggested other music they might like.

GroupLens and Ringo both used a simple collaborative algorithm known as a "user-user" algorithm. This type of algorithm computes the "distance" between pairs of users based on how much they agree on items they have both rated. For instance, if Jim and Jane each give the movie *Tron* five stars, their distance is zero. If Jim then gives *Tron: Legacy* five stars, while Jane rates it three stars, their distance increases. Users whose tastes are relatively "near" each other according to these calculations are said to share a "neighborhood."

But the user-user approach doesn't work that well. For one thing, it's not always easy to form neighborhoods that make sense: Many pairs of users have only a few ratings in common or none at all, and in the case of movies, these few ratings in common tend to be of blockbusters that nearly everyone likes. Also, because the distance between users can change rapidly, user-user algorithms have to do most of their calculations on the spot, and that can take more time than someone clicking around a website is going to hang around.

So most recommenders today rely on an "item-item" algorithm, which calculates the distance between each pair of books or movies or what have you according to how closely users who have rated them agree. People who like books by Tom Clancy are likely to rate books by Clive Cussler highly, so books by Clancy and Cussler are in the same neighborhood. Distances between pairs of items, which may be based on the ratings of thousands or millions of users, tend to be relatively stable over time, so

recommenders can precompute distances and generate recommendations more quickly. Both Amazon and Netflix have said publicly that they use variants of an item-item algorithm, though they keep the details secret.

One problem with both user-user and item-item algorithms is the inconsistency of ratings. Users often do not rate the same item the same way if offered the chance to rate it again. Tastes change, moods change, memories fade. MIT conducted one study in the late 1990s that showed an average change of one point on a seven-point scale a year after a user's original rating. Researchers are trying different ways to incorporate such variables into their models; for example, some recommenders will ask users to rerate items when their original ratings seem out of sync with everything else the recommender knows about them.

But the user-user and item-item algorithms have a bigger problem than consistency: They're too rigid. That is, they can spot people who prefer the same item but then miss potential pairs who prefer very similar items. Let's say you're a fan of Monet's water lily paintings. Of the 250 or so paintings of water lilies that the French impressionist did, which is your favorite? Among a group of Monet fans, each person may like a different water lily painting best, but the basic algorithms might not recognize their shared taste for Monet.

About a decade ago, researchers figured out a way to factor in such sets of similar items—a process called dimensionality reduction. This method is much more computationally intensive than the user-user and item-item algorithms, so its adoption has been slower. But as computers have gotten faster and cheaper, it has been gaining ground.

To understand how dimensionality reduction works, let's consider your taste in food and how it compares with that of a million other people. You can represent those tastes in a huge matrix, where each person's taste makes up its own row and each of the thousands of columns is a different food. Your row might show that you gave grilled filet mignon five stars, braised short ribs four and a half stars, fried chicken wings two stars, cold tofu rolls one star, roasted portobello mushroom five stars, steamed edamame with sea salt four stars, and so forth.

A recommender using the matrix wouldn't really care about your particular rating of a particular food, however. Instead, it wants to understand your preferences in general terms, so that it can apply this knowledge to a wide variety of foods. For instance, given the above, the recommender might conclude that you like beef, salty things, and grilled dishes, dislike chicken and anything fried, are neutral on vegetables, and so on. The number of such taste attributes or dimensions would be much smaller than the number of possible foods—there might be 50 or 100 dimensions in all. And by looking at those dimensions, a recommender could quickly determine whether you'd like a new food—say, salt-crusted prime rib—by comparing its dimensions (salty, beef, not chicken, not fried, not vegetable, not grilled) against your profile. This more general representation allows the recommender to spot users who prefer similar yet distinct items. And it substantially compresses the matrix, making the recommender more efficient.

It's a pretty cool solution. But how do you find those taste dimensions? Not by asking a chef. Instead, these systems use a mathematical technique called singular value decomposition to compute the dimensions. The technique involves factoring the original giant matrix into two "taste matrices"—one

that includes all the users and the 100 taste dimensions and another that includes all the foods and the 100 taste dimensions—plus a third matrix that, when multiplied by either of the other two, re-creates the original matrix.

Unlike the food example above, the dimensions that get computed are neither describable nor intuitive; they are pure abstract values, and try as you might, you'll never identify one that represents, say, "salty." And that's okay, as long as those values ultimately yield accurate recommendations. The main drawback to this approach is that the time it takes to factor the matrix grows quickly with the number of customers and products—a matrix of 250 million customers and 10 million products would take 1 billion times as long to factor as a matrix of 250 000 customers and 10 000 products. And the process needs to be repeated frequently. The matrix starts to grow stale as soon as new ratings are received; at a company like Amazon, that happens every second. Fortunately, even a slightly stale matrix works reasonably well. And researchers have been devising new algorithms that provide good approximations to singular value decomposition with substantially faster calculation times.

By now, you have a basic idea of how an online retailer sizes you up and tries to match your tastes to those of others whenever you shop at its site. Recommenders have two other features that dramatically affect the recommendations you see: First, beyond figuring out how similar you are to other shoppers, the recommender has to figure out what you actually like. Second, the system operates according to a set of business rules that help ensure its recommendations are both helpful to you and profitable for the retailer.

For example, consider the recommender used for Amazon's online art store, which at last count had more than 9 million prints and posters for sale. Amazon's art store (http://www.amazon.com/Art-com/b?ie=UTF8&node=3489231) assesses your preferences in a few ways. It asks you to rate particular artworks on a five-star scale, and it also notes which paintings you enlarge, which you look at multiple times, which you place on a wish list, and which you actually buy. It also tracks which paintings are on your screen at the time as well as others you look at during your session. The retailer uses the path you've traveled through its website—the pages you've viewed and items you've clicked on—to suggest complementary works, and it combines your purchase data with your ratings to build a profile of your long-term preferences.

Companies like Amazon collect an immense amount of data like this about their customers. Nearly any action taken while you are logged in is stored for future use. Thanks to browser cookies, companies can even maintain records on anonymous shoppers, eventually linking the data to a customer profile when the anonymous shopper creates an account or signs in. This explosion of data collection is not unique to online vendors—

(/img/recommendf2-1348086297597.jpg)
Recommendation odyssey: An Amazon user interested in 2001: A Space Odyssey sees suggestions from three different collaborative recommenders.

Click on the image for the full illustration view.

Walmart is famous for its extensive mining of cash register receipt data. But an online shop is much better positioned to view and record not just your purchases but what items you considered, looked at, and rejected. Throughout much of the world, all of this activity is fair game; only in Europe do data privacy laws restrict such practices to a degree.

Of course, regardless of the law, any customer will react badly if his or her data is used inappropriately. Amazon learned this lesson the hard way back in September 2000, when certain customers discovered they were being quoted higher prices because the website had identified them as regular customers, rather than as shoppers who had entered anonymously or from a comparison-shopping site. Amazon claimed this was just a random price test and the observed relationship to being a regular customer was coincidental, but it nevertheless stopped the practice.

The business rules around these systems are designed to prevent recommenders from making foolish suggestions and also to help online retailers maximize sales without losing your trust. At their most basic level, these systems avoid what's known as the supermarket paradox. For example, nearly everyone who walks into a supermarket likes bananas and will often buy some. So shouldn't the recommender simply recommend bananas to every customer? The answer is no, because it wouldn't help the customer, and it wouldn't increase banana sales. So a smart supermarket recommender will always include a rule to explicitly exclude recommending bananas.

That example may sound simplistic, but in one of our early experiences, our system kept recommending the Beatles' "White Album" to nearly every visitor. Statistically this was a great recommendation: The customers had never purchased this item from the e-commerce site, and most customers rated it highly. And yet the recommendation was useless. Everyone who was interested in the "White Album" already owned a copy.

Most recommender rules are more subtle, of course. When John recently searched for an action movie on Netflix, for instance, he wasn't offered *The Avengers*, because the blockbuster was not yet available for rental, and so the suggestion wouldn't have profited Netflix. Instead it steered him to *Iron Man 2*, which was available for streaming.

Other business rules prevent recommenders from suggesting loss leaders—products that sell below cost to draw people into the site—or conversely encourage them to recommend products that are overstocked. During our time at Net Perceptions, we worked with a client who did just that: He used his recommender system to identify—with considerable success—potential customers for his overstocked goods.

This kind of thing quickly gets tricky, however. A system that simply pushes high-margin products isn't going to earn the customers' trust. It's like going to a restaurant where the waiter steers you toward a particular fish dish. Is it really his favorite? Or did the chef urge the staff to push out the fish before its sell-by date?

To build trust, the more sophisticated recommender systems strive for some degree of transparency by giving customers an idea of why a particular item was recommended and letting them correct their profiles if they don't like the recommendations they're getting.

You can, for instance, delete information from your Amazon profile about things you purchased as gifts; after all, those don't reflect your tastes. You can also find out why certain products have been offered through the recommender. After Amazon selected Jonathan Franzen's novel *Freedom* for John, he clicked on the link labeled "Explain." He then got a brief explanation that certain books on John's Amazon wish list had triggered the recommendation. But as John hadn't read any of the wish list books, he discounted the *Freedom* suggestion. Explanations like these let users know how reliable a given recommendation is.

But profile adjustments and explanations often aren't enough to keep a system on track. Recently Amazon bombarded Joe with e-mails for large-screen HDTVs—as many as three a week for months. Besides sending him more e-mail on the topic than he could possibly want, the retailer didn't recognize that he'd already purchased a TV through his wife's account. What's more, the e-mails did not offer an obvious way for Joe to say, "Thanks, but I'm not interested." Eventually, Joe unsubscribed from certain Amazon e-mails; he doesn't miss the messages, and he has more time to actually watch that TV.

So how well do recommenders ultimately work? They certainly are increasing online sales; analyst Jack Aaronson of the Aaronson Group estimates that investments in recommenders bring in returns of 10 to 30 percent, thanks to the increased sales they drive. And they still have a long way to go.

Right now the biggest challenge for those of us who study recommender systems is to figure out how best to judge the new approaches and algorithms. It's not as simple as benchmarking a microprocessor, because different recommenders have very different goals.

The easiest way to evaluate an algorithm is to look at the difference between its predictions and the actual ratings users give. For instance, if John gives the teen-romance novel *Twilight* one star, Amazon might note that it had predicted he would give it two stars, based on the ratings of other similar users, and so its recommender was off by a star. But sellers care much more about errors on highly rated items than errors on low-rated items, because the highly rated items are the ones users are more likely to buy; John is never going to purchase *Twilight*, so scoring this rating contributes little to understanding how well the recommender works.

Another common measure is the extent to which recommendations match actual purchases. This analysis can also be misleading, however, because it erroneously rewards the recommender for items users managed to find on their own—precisely the items they don't need recommendations for!

Given the shortcomings of these approaches, researchers have been working on new metrics that look not just at accuracy but also at other attributes, such as serendipity and diversity.

Serendipity rewards unusual recommendations, particularly those that are valuable to one user but not as valuable to other similar users. An algorithm tuned to serendipity would note that the "White Album" appears to be a good recommendation for nearly everyone and would therefore look for a recommendation that's less common—perhaps Joan Armatrading's *Love and Affection*. This less-popular recommendation wouldn't be as likely to hit its target, but when it did, it would be a much happier surprise to the user.

Looking at the diversity of a recommender's suggestions is also revealing. For instance, a user who loves Dick Francis mysteries might nevertheless be disappointed to get a list of recommendations all written by Dick Francis. A truly diverse list of recommendations could include books by different authors and in different genres, as well as movies, games, and other products.

Recommender systems research has all sorts of new ground to break, far beyond fine-tuning existing systems. Researchers today are considering to what extent a recommender should help users explore parts of a site's collection they haven't looked into—say, sending book buyers over to Amazon's clothing department rather than recommending safe items they may be more comfortable with. Going beyond the retail world, recommenders could help expose people to new ideas; even if we disagree with some of them, the overall effect might be positive in that it would help reduce the balkanization of society. Whether recommenders can do that without annoying us or making us distrustful remains to be seen.

But one thing is clear: Recommender systems are only going to get better, collect more data about you, and show up in new and surprising places. And as for you, if you liked this article, Amazon will be happy to recommend entire books on recommender systems that you might also like.

This article originally appeared in print as "Recommended for You."

About the Authors

Joseph A. Konstan and John Riedl (http://www.grouplens.org/team) ARE both professors of computer science at the University of Minnesota. As codirectors of GroupLens Research, Konstan, an IEEE Senior Member, and Riedl, an IEEE Fellow, helped create the MovieLens (http://movielens.umn.edu/login) recommender system. The pair's scariest recommender moment came during an interview on "ABC Nightline." Just before a station break, MovieLens pitched the 1950s film noir Sunset Boulevard to host Robert Krulwich. Konstan and Riedl had to wait until they were back on the air to hear his verdict on the suggestion: He loved it!