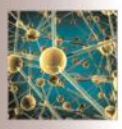
The Test of Time



Two Decades of Recommender Systems at Amazon.com

Amazon is well-known for personalization and recommendations, which help customers discover items they might otherwise not have found. In this update to our original article, we discuss some of the changes as Amazon has grown.

Brent Smith
Amazon.com

Greg Linden
Microsoft

For two decades now,[1] Amazon.com has been building a store for every customer. Each person who comes to Amazon.com sees it differently, because it's individually personalized based on their interests. It's as if you walked into a store and the shelves started rearranging themselves, with what you might want moving to the front, and what you're unlikely to be interested in shuffling further away.

From a catalog of hundreds of millions of items, Amazon.com's recommendations pick a small number of items you might enjoy based on your current context and your past behavior. The algorithms aren't magic; they simply share with you what other people have already discovered. The algorithm does all the work. It's computers helping people help other people, implicitly and anonymously.

Amazon.com launched item-based collaborative filtering in 1998, enabling recommendations at a previously unseen scale for millions of customers and a catalog of millions of items. Since we wrote about the algorithm in *IEEE Internet Computing* in 2003,[2] it has seen widespread use across the Web, including YouTube, Netflix, and many others. The algorithm's success has been from its simplicity, scalability, and often surprising and useful recommendations, as well as desirable properties such as updating immediately based on new information about a customer and being able to explain why it recommended something in a way that's easily understandable.

What was described in our 2003 *IEEE Internet Computing* article has faced many challenges and seen much development over the years. Here, we describe some of the updates, improvements, and adaptations for item-based collaborative filtering, and offer our view on what the future holds for collaborative filtering, recommender systems, and personalization.

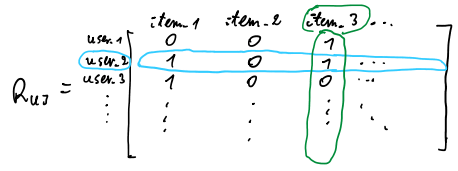
The Algorithm

As we described it in 2003, the item-based collaborative filtering algorithm is straightforward. In the mid-1990s, collaborative filtering was generally user-based, meaning the first step of the algorithm was to search across other users to find people with similar interests (such as similar purchase patterns), then look at what items those similar users found that you haven't found yet. Instead, our algorithm begins by finding related items for each item in the catalog. The term "related" could have several meanings here, but at this point,

Published by the IEEE Computer Society 1089-7801/17/$33.00 © 2017 IEEE

Collaborative filtering

Recommender based solely on user-item interaction



User-based CF

$u_2 = [1, 0, 1, 0, 0, 1, 0 \ldots]$
$u_7 = [1, 0, 1, 0, 0, 0, 0 \ldots]$
$+ \quad + \quad + \quad + \quad - \quad +$

Based on similarity of users

Item-based CF

Based on similarity of items

Standing the Test of Time

As part of recognizing IEEE Internet Computing for its 20 years in publication, I recommended to the editorial board that we pick one of our magazine articles that, over the past 20 years, has withstood the "test of time." In selecting an article, we evaluated the ideas in more than 20 candidate articles that reported on "evergreen" research areas over the past two decades and then assessed these articles based on downloads from IEEE Xplore, citations, and mentions of the work in popular press. This information was presented to a committee consisting of previous Editors in Chief for the magazine. I would like to thank the selection committee from the editorial board — led by Arun Iyengar, and including Fred Douglis, Robert Filman, Michael Huhns, Charles Petrie, Michael Rabinovich, and Munindar Singh. This committee deliberated on the top three articles by evaluating each work's previous importance within the context of its sustained importance in the future.

It's my pleasure to recognize the committee's official "Test of Time" winner: an industry article titled "Amazon.com Recommendations: Item-to-Item Collaborative Filtering" by Greg Linden, Brent Smith, and Jeremy York, from the January/February 2003 issue of IC (see doi:10.1109/MIC.2003.1167344). Fourteen years after the publication of this article, it shows 125 downloads from IEEE Xplore in one month, with more than 12,754 downloads since January 2011. The article currently shows 4,258 citations in Google Scholar. I'm delighted that the selection committee recommended an industry article, as it aligns with the magazine's focus of accessibility in academic, research, and industrial populations.

In addition to recognizing the article, we asked the authors to create this retrospective piece discussing research and insights that have transpired since publishing their winning "Test of Time" article, while projecting into the future.

Going forward, the magazine hopes to celebrate a "Test of Time" article every 2–3 years. I hope that you enjoy this retrospective article, and please take a moment to congratulate Greg Linden, Brent Smith, and Jeremy York.

— M. Brian Blake
Editor-in-Chief, IEEE Internet Computing
Provost and Distinguished Professor, Drexel University

based on conditional probabilities P(Y|X) (kind of)

let's loosely define it as "people who buy one item are unusually likely to buy the other." So, for every item i_1, we want every item i_2 that was purchased with unusually high frequency by people who bought i_1.

Once this related items table is built, we can generate recommendations quickly as a series of lookups. For each item that's part of this customer's current context and previous interests, we look up the related items, combine them to yield the most likely items of interest, filter out items already seen or purchased, and then we are left with the items to recommend.

This algorithm has many advantages over the older user-based collaborative filtering. Most importantly, the majority of the computation is done offline — a batch build of the related items — and the computation of the recommendations can be done in real time as a series of lookups. The recommendations are high quality and useful, especially given enough data, and remain competitive in perceived quality even with the newer algorithms created over the last two decades. The algorithm scales to hundreds of millions of users and tens of millions of items without sampling or other techniques that can reduce the quality of the recommendations. The algorithm updates immediately on new information about a person's interests. Finally, the recommendations can be explained in an intuitive way as arising from a list of items the customer remembers purchasing.

In 2003: Amazon.com, Netflix, YouTube, and More

By the time we published in IEEE in 2003, item-based collaborative filtering was widely deployed across Amazon.com. The homepage prominently featured recommendations based on your past purchases and items browsed in the store. Search result pages recommended items related to your search. The shopping cart recommended other items to add to your cart, perhaps impulse buys to bundle in at the last minute, or perhaps complements to what you were already considering. At the end of your order, more recommendations appeared, suggesting items to order later. Using e-mails, browse pages, product detail pages, and more, many pages on Amazon.com had at least some recommended content, starting to approach a store for every customer.

Others have reported using the algorithm, too. In 2010, YouTube reported using it for recommending videos.[3] Many open source and third-party vendors included the algorithm, and it showed up widely in online retail, travel, news, advertising, and more. In the years following, the recommendations were used so extensively by Amazon.com that a Microsoft Research report estimated 30 percent of Amazon.com's page

Basic idea

user history = $\{i_1, i_7, i_{14}\}$; user basket = $\{i_3, i_5\}$

$\Downarrow$

most similar items to i_1
$\{i_5, i_6\}$
$\cup$
most similar items to i_7
$\{i_5, i_{10}\}$
$\cup$
most similar items to i_{14}
$\{i_1, i_9\}$

$\Downarrow$

all similar items
$\{i_1, i_5, i_6, i_9, i_{10}\}$

$\Downarrow$

remove already bought
$\{i_5, i_6, i_9, i_{10}\}$

$\Downarrow$

identical procedure

The Test of Time

A Present-Day Perspective on Recommendation and Collaborative Filtering

A Present-Day Perspective on Recommendation and Collaborative Filtering

As a PhD student who uses collaborative filtering in my work to introduce customized recommendation techniques (and collaborative filtering) that select "workers" for crowdsourcing,[1,2] the Test of Time article is particularly meaningful to me. Collaborative filtering is a technique used to personalize the experience of users through recommendations tailored to the users' interests, leveraging the experiences of other users with similar profiles. Traditionally, the technique is used in e-commerce platforms to drive sales by converting targeted suggestions to purchases.[3] The technique has rendered more favorable results than blanket advertisement, and is more purposeful toward customizing the experience of individual users. Despite this success, two primary challenges have surfaced: these are concerns related to real-time scalability and recommendation quality. These concerns directly impact the users' individual experiences and, by induction, the success of the platforms using the technique.

The first concern of scalability is directly affected by today's inexpensive and evolving storage and computing capabilities; these have led to overwhelming data generation and collection. Unfortunately, algorithms — including traditional collaborative filtering — haven't evolved in capacity to handle this new volume of data in real time or in an online modality. To address the issue of scalability, a variety of techniques are employed to reduce the dataset in a structured manner. Some of these approaches include sampling users, data partitioning driven by the classification of items, and omitting high- or low-frequency items to bubble others to the top of the recommended list. These approaches, while seeking to remedy the issue in scalability, affect the quality in recommendations; this is a direct impact to the second concern.

Given these concerns, it was incumbent on the research community and practitioners to devise an approach that gains the benefits of scalability without sacrificing recommendation quality. The most successfully employed approach has come in the form of item-based collaborative filtering. Its continued success is evident in applications such as the major large-scale e-commerce platform, Amazon.com. It scopes recommendations via the user's purchased or rated items, pairing them to similar items against established metrics, and finally composing a list of similar items as recommendations. As opposed to dataset-reduction techniques employed through user-centric means, this approach is item-centric, which drastically reduces the data space for evaluation. As outlined in *IEEE Internet Computing*'s Test of Time article[4] and other closely related work,[5] this data-space reduction is potentially up to three orders of magnitude of its original size. Being item-centric, it overcomes the issue with sparsity in user data in traditional approaches (such approaches contribute largely to unnecessary evaluation). It also overcomes the issue of the density in frequent users who have large portions of data associated with their profiles.

Item-based collaborative filtering still requires offline processing to pair similar items. By preprocessing this information offline, recommendations in the list produced from item-based collaborative filtering can occur in real time in an online modality. This allows for easy, quick, more personalized recommendation for the user. The similar items list is a sleek subset of items targeted to the user's purchasing or rating history, as opposed to that of others in the entire dataset. It also overcomes challenges with newer and less-frequent users with sparse history, because the similar items list focuses on the user's history as opposed to the history of other users. This technique is more efficient, yet it hasn't had any adverse effects on the quality of recommendations; as such, it continues to be the technique of choice for real-time, online collaborative filtering and recommendations.

— Julian Jarrett
PhD Student, Computer Science, Drexel University

References

1. J. Jarrett and M.B. Blake, "Using Collaborative Filtering to Automate Worker-Job Recommendations for Crowdsourcing Services," *Proc. 2016 IEEE Int'l Conf. Web Services*, 2016, pp. 641–645.
2. J. Jarrett et al., "Self-Generating a Labor Force for Crowdsourcing: Is Worker Confidence a Predictor of Quality?" *Proc. 2015 3rd IEEE Workshop on Hot Topics in Web Systems and Technologies*, 2015, pp. 85–90.
3. J.L. Herlocker, et al., "Evaluating Collaborative Filtering Recommender Systems," *ACM Trans. Information Systems*, vol. 22, no. 1, 2004, pp. 5–53.
4. G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, vol. 7, no. 1, 2003, pp. 76–80.
5. B. Sarwar et al., "Item-Based Collaborative Filtering Recommendation Algorithms," *Proc. 10th Int'l Conf. World Wide Web*, 2001, pp. 285–295.

views were from recommendations.[4] Similarly, Netflix used recommender systems so extensively that their Chief Product Officer, Neil Hunt, indicated that more than 80 percent of movies watched on Netflix came through recommendations,[5] and placed the value of Netflix recommendations at more than US$1 billion per year.

When we originally developed item-based collaborative filtering, Amazon.com was primarily a bookstore. Since then, Amazon.com's sales have grown more than a hundred-fold and have expanded beyond books to be dominated by non-media items, from laptop computers to women's dresses. This growth challenged many

$$E_{XY} = \sum_{c \in X} \left|1 - (1 - P_Y)^{|c|}\right| = \sum_{c \in X} \left|1 - \sum_{k=0}^{|c|} \binom{|c|}{k} |-P_Y|^k\right|$$

$$= \sum_{c \in X} \left|1 - \left|1 + \sum_{k=1}^{|c|} \binom{|c|}{k} |-P_Y|^k\right|\right| = \sum_{c \in X} \sum_{k=1}^{|c|} (-1)^{k+1} \binom{|c|}{k} P_Y^k$$

$$= \sum_{c \in X} \sum_{k=1}^{\infty} (-1)^{k+1} \binom{|c|}{k} P_Y^k \qquad \text{(since } \binom{|c|}{k} = 0 \text{ for } k > |c|\text{)}$$

$$= \sum_{k=1}^{\infty} \sum_{c \in X} (-1)^{k+1} \binom{|c|}{k} P_Y^k \qquad \text{(Fubini's theorem)}$$

$$= \sum_{k=1}^{\infty} c_k[X] P_Y^k \qquad \text{where } c_k[X] = \sum_{c \in X} (-1)^{k+1} \binom{|c|}{k}.$$

$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$

little space needed to save all data

Figure 1. The derivation of the expected number of customers who bought both items X and Y, accounting for multiple opportunities for each X-buyer to buy Y.

assumptions in our original algorithms, requiring adaptation to a new and changing landscape. Through experience, we also found ways to refine the algorithm to produce more relevant recommendations for the many new applications of it.

Defining "Related" Items

The quality of recommendations depends heavily on what we mean by "related." For example, what do we mean by "unusually likely" to buy item Y given that you bought X? When we observe that customers have bought both X and Y, we might wonder how many X-buyers would have randomly bought Y if the two items were unrelated. A recommender system is ultimately an application of statistics. Human behavior is noisy, and the challenge is to discover useful patterns among the randomness.

A natural way to estimate the number of customers N_{XY} who have bought both X and Y would be to assume X-buyers had the same probability, $P(Y) = |Y \text{ buyers}|/|\text{all buyers}|$, of buying Y as the general population and use $|X \text{ buyers}| * P(Y)$ as our estimate, E_{XY}, of the expected number of customers who bought both X and Y. Our 2003 article, and much of our work before 2003, used a calculation similar to this.

However, it's a curious fact that, for almost any two items X and Y, customers who bought X will be much more likely to buy Y than the general population. How can that be? Imagine a heavy buyer — someone who has bought every item in the catalog. When we look for all the customers who have bought X, this customer is guaranteed to be selected. Similarly, a customer who has made 1,000 purchases will be about 50 times as likely to be selected as someone with 20 purchases; sampling a random purchase doesn't give a uniform probability of selecting customers. So, we have a biased sample. For any item X, customers who bought X will be likely to have bought more than the general population.

This non-uniform distribution of customer purchase histories means we can't ignore who bought X when we're trying to estimate how many X-buyers we would expect to randomly buy Y. We found it useful to model customers as having many chances to buy Y.[6] For example, for a customer with 20 purchases, we take each of these 20 purchases as an independent opportunity to have purchased Y.

More formally, for a given customer c who purchased X [denoted by $c \in X$], we can estimate c's probability of buying Y as $1 - (1 - P_Y)^{|c|}$, where $|c|$ represents the number of non-X purchases made by c and $P_Y = |Y \text{ purchases}|/|\text{all purchases}|$ or the probability that any randomly selected purchase is Y. Then, we can calculate the expected number of Y-buyers among the X-buyers by summing over all X-buyers and using a binomial expansion (see Figure 1).

We can write E_{XY} as a polynomial in P_Y with coefficients that depend purely on X. In practice, P_Y's are small, so close approximations can be made with bounded k. In addition, P_Y and $\alpha_k[X]$ can be precomputed for all items, which

ally likely to buy a certain memory card, but this doesn't guarantee that the memory card works with the camera. Customers buy memory cards for many reasons and the observed correlation might be a random occurrence. Indeed,

---

{'1, '5, '6, '9, '10}

⇓ remove already bought

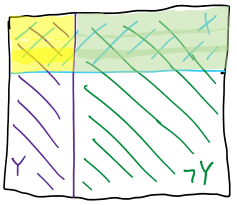
{i5, i6, i9, i10}

⇓ order by similarity and recommend in that order

Simple approach to estimate E_{XY} (used before 2003)

We assume that:
- every user buys either 0 or 1 product X and the same holds for Y
- users are independent of each other
- items are independent of each other

Then the number of users who bought X (or Y) is governed by the Bernoulli distribution $\mathcal{B}(N, P(X))$ where N is the number of all users and P(X) (or P(Y)) is the probability of buying X (or Y). As buying X and Y is independent, then the number of users who bought both X and Y is governed by Bernoulli distribution $\mathcal{B}(N, P(X) \cdot P(Y))$



For independent events $P(X \cap Y) = P(X) \cdot P(Y)$

In general $P(Y|X) = \frac{P(Y \cap X)}{P(X)}$

$P(\neg Y|X) = \frac{P(\neg Y \cap X)}{P(X)}$

Hence

$$E_{XY} = \mathbb{E}\left(\mathcal{B}(N, P(X) \cdot P(Y))\right) = N \cdot P(X) \cdot P(Y)$$

Our best guess from data about P_X and P_Y is

$$P_X = \frac{|X \text{ buyers}|}{N} \qquad P_Y = \frac{|Y \text{ buyers}|}{N}$$

Then

$$E_{XY} = |X \text{ buyers}| \cdot \frac{|Y \text{ buyers}|}{N} = |X \text{ buyers}| \cdot P(Y)$$

Idea: Compare N_{XY} with $E_{XY} = \mathbb{E}[Y|X] = \mathbb{E}[N_{XY}]$ based on joint probability distribution assuming that all items are unrelated

Proved to be too simplistic!

Solution: take the number of purchases for each customer into account

We can write E_{XY} as a polynomial in P_Y with coefficients that depend purely on X. In practice, P_Y's are small, so close approximations can be made with bounded k. In addition, P_Y and $\alpha_k(X)$ can be precomputed for all items, which then allows E_{XY} to be approximated for any pair of items with a simple combination of precomputed values.

With a robust method of computing E_{XY}, we can use it to evaluate whether N_{XY}, the observed number of customers who bought both X and Y, is higher or lower than randomly would be expected. For example, $N_{XY} - E_{XY}$ gives an estimate of the number of non-random co-occurrences, and $[N_{XY} - E_{XY}]/E_{XY}$ gives the percent difference from the expected random co-occurrence. These are two examples of creating a similarity score $S[X, Y]$ as a function of the observed and expected number of customers who purchased both X and Y. The first, $N_{XY} - E_{XY}$, will be biased toward popular Y's such as the first Harry Potter book, so the recommendations might be perceived as too obvious or irrelevant. The second, $[N_{XY} - E_{XY}]/E_{XY}$, makes it too easy for low-selling items to have high scores, so the recommendations might be perceived as obscure and random, especially because of the large number of unpopular items. Relatedness scores need to strike a balance between popularity on one end and the power law distribution of unpopular items on the other. The chi-square score, $[N_{XY} - E_{XY}]/\sqrt{E_{XY}}$, is an example that strikes such a balance.

There are several other choices and parameters that could be considered in a relatedness score and in creating recommendations from related items. Our experience is that there is no one score that works best in all settings. Ultimately, perceived quality is what recommendations are judged on; recommendations are useful when people find them useful.

Machine learning and controlled online experimentation can learn what customers actually prefer, picking the best parameters for the specific use of the recommendations. Not only can we measure which recommendations are effective, but we can also feed information about which recommendations people liked, clicked on, and bought back into our algorithms, learning what helps customers the most.[7]

For example, compatibility is an important relationship. We might observe that customers who buy a particular digital camera are unusually likely to buy a certain memory card, but this doesn't guarantee that the memory card works with the camera. Customers buy memory cards for many reasons and the observed correlation might be a random occurrence. Indeed, there are hundreds of thousands of memory cards in Amazon.com's catalog, so many of them are randomly correlated with the camera. Many e-commerce sites use a hand-curated knowledge base of compatibility, which is expensive and error-prone to maintain, especially at Amazon.com's scale. We found that, given enough data and a robust metric for the relatedness of items, compatibility can emerge from people's behavior, with the false signals failing away and the truly appropriate items surfacing.

Curiously, we found that the meaning of related items also can be emergent, arising from the data, and discovered by customers themselves. Consider the items people look at versus the items they purchase. For books, music, and other low-cost items, people tend to look at and purchase the same thing. For many expensive items, and especially for non-media items, what people view and what they purchase can be radically different. For example, people tend to look at many televisions, but only purchase one. What they look at around the time of looking at that television will tend to be other televisions. What they purchase around the time they bought a television tends to be complements that enhance the experience after buying that particular television, such as a Blu-ray player and a wall mount.

The Importance of Time

Understanding the role of time is important for improving the quality of recommendations. For example, when computing the related items table, how related a purchase is to another purchase depends heavily on their proximity in time. If a customer buys a book five months after buying another book, this is weaker evidence for the books being related than if the customer had purchased them on the same day. Time directionality also can be helpful. For example, the fact that customers tend to buy a memory card after buying a camera, rather than the other way around, might be a good hint that we shouldn't recommend the camera when someone buys the memory card. Sometimes, items are bought sequentially, such as a book, movie, or TV series,

and recommendations should be for what you want to do next.

Amazon.com's catalog is continually changing through time. Every day, thousands of new items arrive and many others fade into obscurity and obsolescence. This cycle is especially pronounced in some categories. For example, apparel has seasonal fashions, and consumer electronics has rapid technological innovation. New items can be at a disadvantage, because they don't have enough data yet to have a strong correlation with other items. This is referred to as the cold-start problem, and often requires an explore/exploit process to give items that have not yet had much opportunity to be purchased an opportunity to be shown. Perishable items such as news or social media posts represent a particularly challenging form of cold start, often requiring blending data from content-based algorithms (using subject, topic, and text) with behavior-based algorithms (using purchases, views, or ratings).

Customers also have a lifecycle and experience their own cold-start problem. Knowing what to recommend when we have very limited information about a new customer's interests has long been an issue. When to make use of limited information and when to play it safe with generally popular items is a subtle transition that's difficult to get correct.

Even for established customers, modeling time correctly has a large impact on the quality of recommendations. As they age, previous purchases become less relevant to the customer's current interests. This is complicated by the fact that this relevance can attenuate at different rates for different types of items. For example, some purchases – such as a manual on sailing heavy seas – likely indicate a durable long-term interest. Others such as a dishwasher repair kit might not be relevant after this weekend's project. There are even some purchases such as baby rattles where the recommendations have to change over a long period of time; four years later, we should recommend balance bikes and board books rather than baby bottles and teethers. And some items, such as books, are usually only bought once; others, such as toothpaste, are bought again and again with a fairly predictable lapse of time between the purchases.

The quality of recommendations we can make depends not only on the timing of past purchases, but what was purchased. We found that a single book purchase can say a lot about a customer's interests, letting us recommend dozens of highly relevant items. But, many purchases in non-media categories tell us little about the customer. What insights can be gleaned from the purchase of a stapler? What surprising and insightful recommendations can be made from buying a pair of socks? Recommending tape dispensers or more underwear might be helpful in the moment, but leads to uninspiring recommendations in the longer term. Thus, we had to develop techniques for learning which purchases lead to useful recommendations and when some should be ignored.

Finally, the importance of diversity in recommendations is well known; sometimes it's better to give a variety of related items rather than a narrowly targeted list. The breadth of Amazon.com's massive catalog with its many types of products offers a unique challenge not seen in single-product category stores such as bookstores. For example, recommending more books to a heavy reader might lead to a sale, but people might benefit most long term by discovering items they have never even considered before in another product line. Immediate intent is a factor in diversity as well. When someone is clearly seeking something specific, recommendations should be narrow to help them quickly find what they need. But when intent is unclear or uncertain, discovery and serendipity should be the goal. Finding the right balance in the diversity of recommendations requires experimentation along with a willingness to optimize for the long term.

The Future: Recommendations Everywhere

What does the future hold for recommendations? We believe there's more opportunity ahead of us than behind us. We imagine intelligent interactive services where shopping is as easy as a conversation.

This moves beyond the current paradigm of typing search keywords in a box and navigating a website. Instead, discovery should be like talking with a friend who knows you, knows what you like, works with you at every step, and anticipates your needs.

This is a vision where intelligence is everywhere. Every interaction should reflect who you are and what you like, and help you find what

E_{XY} ...

Proved to be too simplistic !

Solution : take the number of purchases for each customer into account

$$P(Y \mid c \in X) = 1 - (1 - P(Y))^{|c|}$$

$|c|$ - number of non-X purchases

$$E_{XY} = \sum_{c \in X} \left(1 - (1 - P(Y))^{|c|}\right)$$

vs previous model

$$E_{XY} = |X| \cdot P(Y) = \sum_{c \in X} P(Y)$$

Importance of proper mathematical modeling !

other people like you have already discovered. It should feel hollow and pathetic when you see something that's obviously not you; do you not know me by now?

Getting to this point requires a new way of thinking about recommendations. There shouldn't be recommendation features and recommendation engines. Instead, understanding you, others, and what's available should be part of every interaction.

Recommendations and personalization live in the sea of data we all create as we move through the world, including what we find, what we discover, and what we love. We're convinced the future of recommendations will further build on intelligent computer algorithms leveraging collective human intelligence. The future will continue to be computers helping people help other people.

Nearly two decades ago, Amazon.com launched recommendations to millions of customers over millions of items, helping people discover what they might not have found on their own. Since then, the original algorithm has spread over most of the Web, been tweaked to help people find videos to watch or news to read, been challenged by other algorithms and other techniques, and been adapted to improve diversity and discovery, recency, time-sensitive or sequential items, and many other problems. Because of its simplicity, scalability, explainability, adaptability, and relatively high-quality recommendations, item-based collaborative filtering remains one of the most popular recommendation algorithms today.

Yet the field remains wide open. An experience for every customer is a vision none have fully realized. Much opportunity remains to add intelligence and personalization to every part of every system, creating experiences that seem like a friend that knows you, what you like, and what others like, and understands what options are out there for you. Recommendations are discovery, offering surprise and delight with what they help uncover for you. Every interaction should be a recommendation.

References

1. G.D. Linden, J.A. Jacobi, and E.A. Benson, *Collaborative Recommendations Using Item-to-Item Similarity Mappings*, US Patent 6,266,649, to Amazon.com, Patent and Trademark Office, 2001 (filed 1998).
2. G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, vol. 7, no. 1, 2003, pp. 76–80.
3. J. Davidson et al., "The YouTube Video Recommendation System," *Proc. 4th ACM Conf. Recommender Systems*, 2010, pp. 293–296.
4. A. Sharma, J.M. Hofman, D.J. Watts, "Estimating the Causal Impact of Recommendation Systems from Observational Data," *Proc. 16th ACM Conf. Economics and Computation*, 2015, pp. 453–470.
5. C.A. Gomez-Uribe and N. Hunt, "The Netflix Recommender System: Algorithms, Business Value, and Innovation," *ACM Trans. Management Information Systems*, vol. 6, no. 4, 2016, pp. 1–19.
6. B. Smith, R. Whitman, and G. Chanda, *System for Detecting Probabilistic Associations between Items*, US Patent 8,239,287, to Amazon.com, Patent and Trademark Office, 2012.
7. K. Chakraborti and B. Smith, *Method and System for Associating Feedback with Recommendation Rules*, US Patent 8,090,621, to Amazon.com, Patent and Trademark Office, 2012.

Brent Smith has worked on personalization and recommendations at Amazon.com for 17 years, leading teams that work on fast-paced customer-facing innovation. Smith has a BS in mathematics from the University of California, San Diego, and an MS in mathematics from the University of Washington. Contact him at smithbr@amazon.com.

Greg Linden is a data scientist at Microsoft (previously at Amazon.com, Google, and several startups). Much of his previous work was in recommendations, personalization, artificial intelligence, search, and advertising. Linden has an MS in computer science from the University of Washington and an MBA from Stanford University. Contact him at glinden@gmail.com.