The Big Promise of Recommender Systems

Francisco J. Martin, Justin Donaldson, Adam Ashenfelter, Marc Torrens, and Rick Hangartner

■ *Recommender systems have been part of the Internet for almost two decades. Dozens of vendors have built recommendation technologies and taken them to market in two waves, roughly aligning with the web 1.0 and 2.0 revolutions. Today recommender systems are found in a multitude of online services. They have been developed using a variety of techniques and user interfaces. They have been nurtured with millions of users' explicit and implicit preferences (most often with their permission). Frequently they provide relevant recommendations that increase the revenue or user engagement of the online services that operate them. However, when we evaluate the current generation of recommender systems from the point of view of the "recommendee," we find that most recommender systems serve the goals of the business instead of their users' interests. Thus we believe that the big promise of recommender systems has yet to be fulfilled. We foresee a third wave of recommender systems that act directly on behalf of their users across a range of domains instead of acting as a sales assistant. We also predict that such new recommender systems will better deal with information overload, take advantage of contextual clues from mobile devices, and utilize the vast information and computation stores available through cloud-computing services to maximize users' long-term goals.*

A recommender system is a software application capable of suggesting interesting things to its users after learning their preferences over time (Jannach et al. 2010, Ricci et al. 2011). Recommender systems were envisioned in the 1970s (Negroponte 1970), conceptualized and prototyped in the early 1990s (Goldberg et al. 1992), and implemented and first commercialized in the mid-1990s (Resnick and Varian 1997). They have two (sometimes diametrically opposed) value propositions that contribute to their popularity. On one hand, they help their users cope with the problem of information overload (that is, they take the users' side). On the other hand, they provide companies that operate them with an effective way to drive more sales or increase the level of engagement of the services that they offer (that is, they take the business's side).

Multiple vendors have taken recommender systems to market in a series of waves over the last two decades. The first two waves can respectively be matched with the web 1.0 and 2.0 revolutions. Both waves introduced dozens of companies selling recommendation technologies (see table 1) and thousands of companies using recommender systems with the final goal of increasing their revenues. Not all of the recommender vendors that started during these waves are still active. Nevertheless, recommender services are now a ubiquitous part of many online services. Dozens of applications capture our explicit and implicit preferences on a daily basis (normally with our permission) with the goal of recommending something interesting in the near future (Kirn 2010). Hundreds of researchers around the world have contributed to improve the accuracy, scalability, security, and many other aspects of recommendation systems (Konstan, Riedl, and Smyth 2007; Pu et al. 2008; Bergman et al. 2009; Amatriain et al. 2010).

However, when we look at the current recommender systems generation from the point of view of the "recommendee" (users' side) we can see that recommender systems are more inclined toward achieving short-term sales and business goals. Instead of helping their users to cope with the problem of information overload they can actually contribute to information overload by proposing recommendations that do not meet the users' current needs or interests. Consider the following questions: Could you imagine the Netflix recommender suggesting that you watch a TV show that is broadcasted tonight instead of prompting you to stream another movie from the Netflix repository? Could you imagine the Amazon recommender suggesting that you borrow a novel from your friend, who already bought it a few months ago, instead of recommending to buy it now? Could you imagine a bank recommender suggesting that you transfer your savings to a certified deposit with higher interest in another bank? Could you imagine the iTunes recommender suggesting that you stop buying because you are exceeding your monthly budget for songs?

The recommendations we just mentioned are hard to imagine because recommender systems have mostly proliferated on the business's side. There is no technical reason why a recommender system could not make such suggestions by considering a broader view of users' goals and sources of information. In fact recommender systems were originally envisioned to operate this way (Goldberg et al. 1992; Maes 1994; Resnick et al. 1994; Negroponte 1995) but most vendors favored licensing their recommendation technology to retailers over direct-to-consumer initiatives. However, the current confluence of easily programmable mobile devices, open APIs, distribution channels (such as the Apple App store), and the low cost of cloud-based computing and storage creates an excellent breeding ground for a new wave of direct-to-consumer recommender systems.

One aim of this article is to encourage both researchers and practitioners to embark in new commercial adventures. The window of opportunity is now open to innovate in a third generation of recommender systems that act directly on behalf of their users and help them cope with information overload. Next we review the first two waves of recommender systems, share lessons learned, and describe what we can expect from the next wave.

The First Wave of Recommender Systems

In the 1980s the increasing use of electronic mail and newsgroups caused the first common information overload problems (Palme 1984). Researchers started to investigate new methods of handling these issues, drawing on theories and techniques from general information filtering systems (Goldberg et al. 1992; Resnick et al. 1994) and word-of-mouth automation (Shardanand and Maes 1995). At this early stage, researchers also investigated more sophisticated systems that worked proactively on behalf of the user in personal computing contexts (Kozierok and Maes 1993). Pattie Maes used the metaphor of a *personal assistant* to exemplify the collaboration between a user and an interface agent capable of learning the user's interests, habits, and preferences to cope with the problem of information and work overload (Maes 1994). As the agent gradually learns how better to assist its user, the range of tasks that can be delegated increases. Common early tasks included scheduling meetings and filtering email.

Researchers reasoned that harnessing the collective assessment of many individuals could quickly establish abstract notions of priority and classification. This basic idea was the basis for Tapestry, the first experimental system supporting *collaborative filtering*. Tapestry was built at the Xerox Palo Alto Research Center to handle the (relatively) large volume of email received by their researchers at the time (Goldberg et al. 1992).

In addition to email, collaborative filtering quickly found success in a number of domains. The GroupLens project used collaborative filtering in order to help people find interesting articles in online newsgroups (Resnick et al. 1994). In 1995, the GroupLens team proposed movie recommendations through the Movielens project (see figure 1). The Ringo system at MIT used social information filtering in order to generate personalized music recommendations (Shardanand and Maes 1995).

It was not long before collaborative filtering techniques found their way into commercial products. In 1995 Firefly Networks (originally Agents, Inc.) became the first company to focus on offering a recommendation service. Firefly Networks used a successor of Ringo that was capable of generating user profiles to recommend music. It expanded its service to recommend movies and websites as well.

As the 1990s ended, more and more commercial applications included collaborative filtering, and efforts were made to better monetize the new tech-

First Wave of Recommender Systems Vendors
Companies founded between 1991 and 2000
ATG Dynamo (acquired by Oracle), Blue Martini (acquired by Escalate Retail), Braisins, Broadvision, E.piphany (acquired by SSA Global Technologies acquired by Infor), Firefly Network (acquired by Microsoft), IBM WebSphere Personalization, Like Minds (acquired by Andromedia), Manna, Open Sesame (acquired by Bowne Inc.), Net Perceptions, Vignette (acquired by Open Text).
Second Wave of Recommender Systems Vendors
Companies founded between 2001 and 2010
Adobe Recommendations (before Omniture recommendations), Aggregate Knowledge, Amadesa, Avail Intelligence, Barilliance, Bay Note, Blue Know, Certona, Changing Worlds (acquired by Amdocs), Choice Stream, Clever Set (acquired by ATG Dynamo), Commendo, Criteo, Directed Edge, Easyrec, 4 Tell, Gravity, Ericsson SDP, I Go Digital, Iletken, Istobe, July Systems, Loomia, Match Mine, Media Unbound (acquired by Rovi), My Buys, Olista (acquired by Connectiva), Pontis, Predictive Intent, Prediggo, Rich Relevance, 7 Billion People, Strands, We Like, Xiam (acquired by Qualcomm), Xtract.

Table 1. Most of the Key Vendors of Recommender Systems Founded in the Last Two Decades.

nology. Resnick and Varian (1997) proposed three business models for independent, direct-to-consumer recommender systems. These included subscriptions or pay-per-use, advertiser support, and content-owner support. However, recommender systems were costly to run and maintain. Therefore, many recommender system startups focused primarily on licensing the technology to the burgeoning e-commerce domain. Net Perceptions, founded by a group of researchers at University of Minnesota in 1996, quickly became the leading vendor of business-to-business recommender systems, powering the recommendations of many of the key web companies. Likeminds, created in 1997 by O'Reilly and AOL, also specialized in commercializing collaborative filtering tools to help other websites offer personalized recommendations. In 1998, Firefly Network was acquired by Microsoft and its technology seeded Microsoft's Passport (now known as LiveID).

As the millennium ended, new web portals and e-commerce sites were in fierce competition to attract users, differentiating their services by actively developing their recommendations. Very soon a number of software manufacturers of application and content management servers (such as E.piphany, Blue Martini, Vignette, and others) complemented their product suites with recommendation or personalization modules. Table 1 gives a simple overview of the relevant businesses. By the end of 1999 many popular websites were already buying recommender system services or were planning to roll out their own service. However, in March 2000 the dot-com bubble burst and the e-commerce domain collapsed, with companies losing more than 5 trillion dollars of market value from March 2000 to October 2002.

Although the views and experiences of the relevant vendors may vary to some extent, there are a number of prominent lessons learned during the first wave that shaped the commercial ventures of the second wave of recommender systems (see table 2).

Figure 1. A Screenshot from Movielens.

This screenshot is an example of a first-wave recommender system for movies and the first high quality source of data for recommender system research. (Used with permission.)

The Second Wave of Recommender Systems

The number of Internet users and the number of websites never stopped growing despite the collapse of the Internet sector. At the end of 2000 there were more than 250 million Internet sub-

Integration Is Much Harder Than It Looks
Schedule enough time upfront for all the last-mile problems you will find. Much more if you try to integrate in a constantly changing environment such as the one many web companies had when initial web technologies were still under development.
Customer's In-House Competition Is Fiercer Than Market Competition
Integrating a recommender system implies parallel battles with the IT and marketing departments in midsize companies. Don't waste your time with big web companies as they see recommendations as too strategic to be delegated.
Cold Start Is An Issue If Your Algorithm Is Exclusively Fed by the System's Own Data
If the recommender system is purely based on your own data, make sure it is started with a substantial initial amount, or wait to unveil the service until enough data is collected. Otherwise the initial experience for the users will be very poor.
Don't Forget About Scalability
Deciding what processes a recommender system must run online and which offline is crucial to be able to scale. The first generation of recommender systems tried to solve everything online or with memory-based approaches that did not scale when the number of users or number of items to recommend increased.

Table 2. Some Lessons Learned During the First Wave of Recommender Systems.

scribers (see figure 2). These users became more eager to share their own content through personal web pages, blogs, and online communities. The amount of user-generated content soon surpassed professional media content, and the social or web 2.0 revolution started. While the market for various digital media content exploded, other problems soon surfaced as user-generated content exhibited some of the same issues encountered in collaborative-filtering-based recommender systems (Su and Khoshgoftaa 2009). In particular, problems related to *scalability*, *sparsity*, and *shilling* became much more challenging to solve.

In 2002, 2003, and 2004, Friendster, MySpace, and Facebook were launched (respectively). Many people started to amass online friends and broadcast each detail of their personal lives. Social networking quickly became a vital personal communication channel as it substituted for email, Short Message Service (SMS), or even face-to-face communication for many people.

As the usage of their systems grew, web companies started to analyze recommendation algorithms in order to make them more scalable, robust, and less prone to bias or sabotage. Amazon.com, Netflix, and Yahoo! Music popularized their recommender systems and solved many of the scalability issues of the first recommender systems (Linden, Smith, and York 2003). Five-star rating schemes combined with purchase and rental behavior patterns became a common source of data for most big e-commerce websites. However, despite the growth and improvements of the underlying algorithms and architecture, recommendation technologies were still too complex or expensive to be affordable by smaller online retailers.

Rating-based collaborative filtering systems relied on their users to rate dozens of items before the service could provide appropriate suggestions. The systems also tended to be static, with no ability to adapt to a given user's changing tastes. This caused problems for domains where many items were unrated, and a user's taste could be expected to change quickly. Music was one such domain, and an important component of the burgeoning digital media market. This opportunity led to the development and adoption of techniques that minimized the number of explicit interactions to train a recommender, similar to the approach introduced by the experimental Ringo system several years prior. Sites like Audio Scrobbler (later merged with Last.fm), Music Mobs, MOG, and MyStrands started tracking what people were listening to on their computers to tacitly indicate interest, and to automatically generate social-based recommendations with little to no prior information required from the user.

Pandora (formerly known as the Music Genome Project) is noteworthy in the way that it integrated a recommender system into its music-streaming service. Instead of using listening behavior or user-generated playlists to relate music, Pandora used dozens of music experts that rated songs following hundreds of preestablished attributes. Then it used a basic rating system (thumbs up, thumbs down) to tailor future recommendations for each user. However, despite operating for nearly 10 years, Pandora still can only offer recommendations for around one million songs. It is currently an important example in the long debate on the trade-off between scalability and the quality of expert-based recommendations, which goes beyond the scope of this article.

In addition to the differences in music recom-

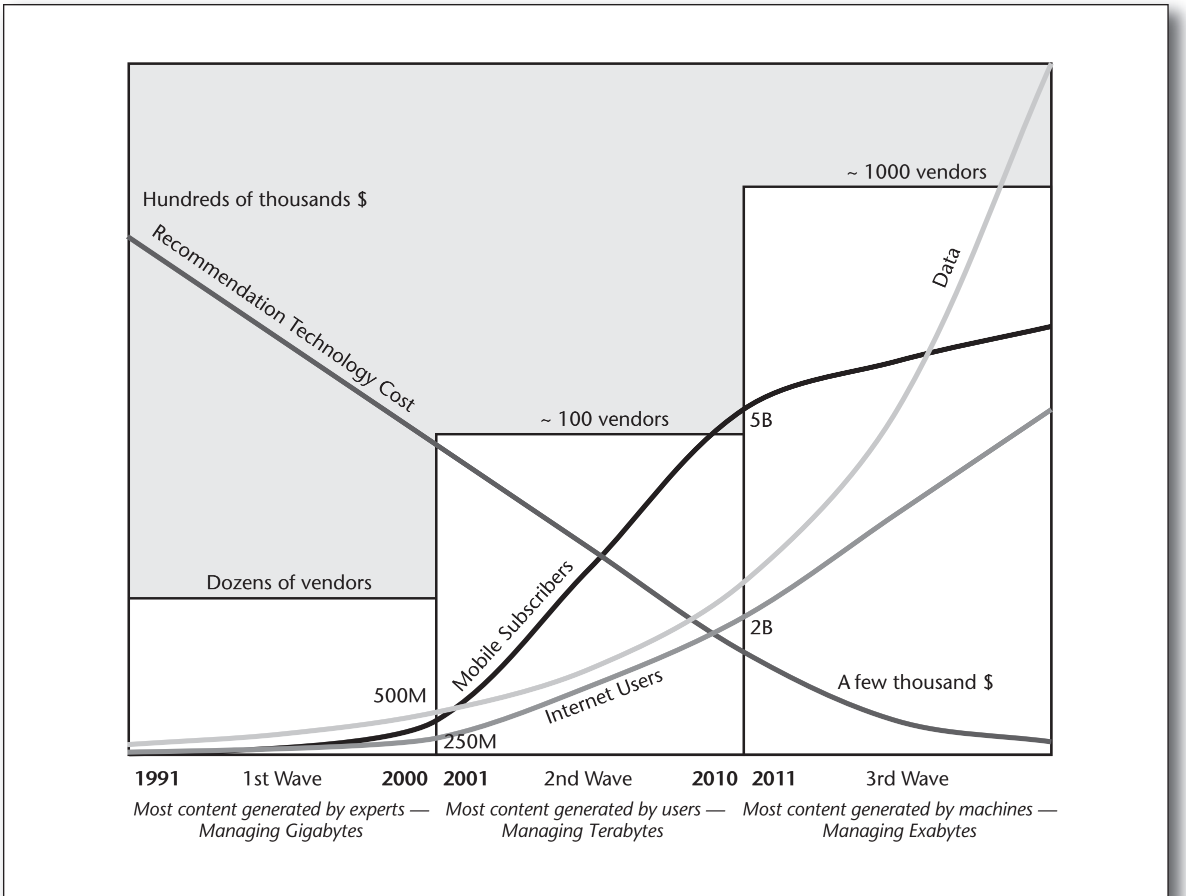

Figure 2. Internet Trends 1991–2010.

mendation approaches, online news also popularized different forms of recommender systems. Initially, Findory offered a personalized news service utilizing content-based collaborative filtering techniques for news story recommendation (figure 3). However, online news sites such as Digg, Reddit, and Hacker News soon became the more popular means for finding relevant news. Instead of more personalized recommender algorithms, stories were promoted and recommended primarily by global popularity trends.

In 2006 Strands organized a summer school on the "Present and Future of Recommender Systems" that brought together researchers, practitioners, and students. Building on its success, the Association for Computing Machinery (ACM) recommender systems conference "RecSys" was established, held in Minneapolis (2007), Lausanne (2008), New York (2009), Barcelona (2010), and Chicago (2011) underway. Collaborative filtering has been the method underlying most recommender systems studies, as can be seen in the weighted tag cloud visualization in figure 4.

While research on algorithms has captured most of the attention for recommender system studies, the rise of social networking indicated that simply keeping tabs on your friends' listening, watching, and reading habits is a more natural way to keep informed of the latest trends. Undirected word-of-mouth suggestions were automatically transmitted through systems like Facebook's News Feed without algorithmic processing but embedded in a convenient interface.

At the end of 2006, Netflix challenged the world to improve the accuracy of its movie recommendation system by 10 percent. It created a contest that brought lots of attention to recommender systems, not just for the $1,000,000 prize, but for the fact that a popular service like Netflix was willing to invest in such an exotic scheme just to improve the quality of its recommender service. In 2009 the BellKor "Pragmatic Chaos" team won the contest (Töscher, Jahrer, and Bell 2009). Their efforts may have set the bar for a best-in-class recommendation algorithm for movies, but practitioners have noted that algorithmic precision is just one of many factors that affect a user's adoption of a recommendation, and other issues such as interface design, long-term performance evaluation, or context-awareness are also prominent parts of a recommender system outcome.

A number of companies saw the opportunity

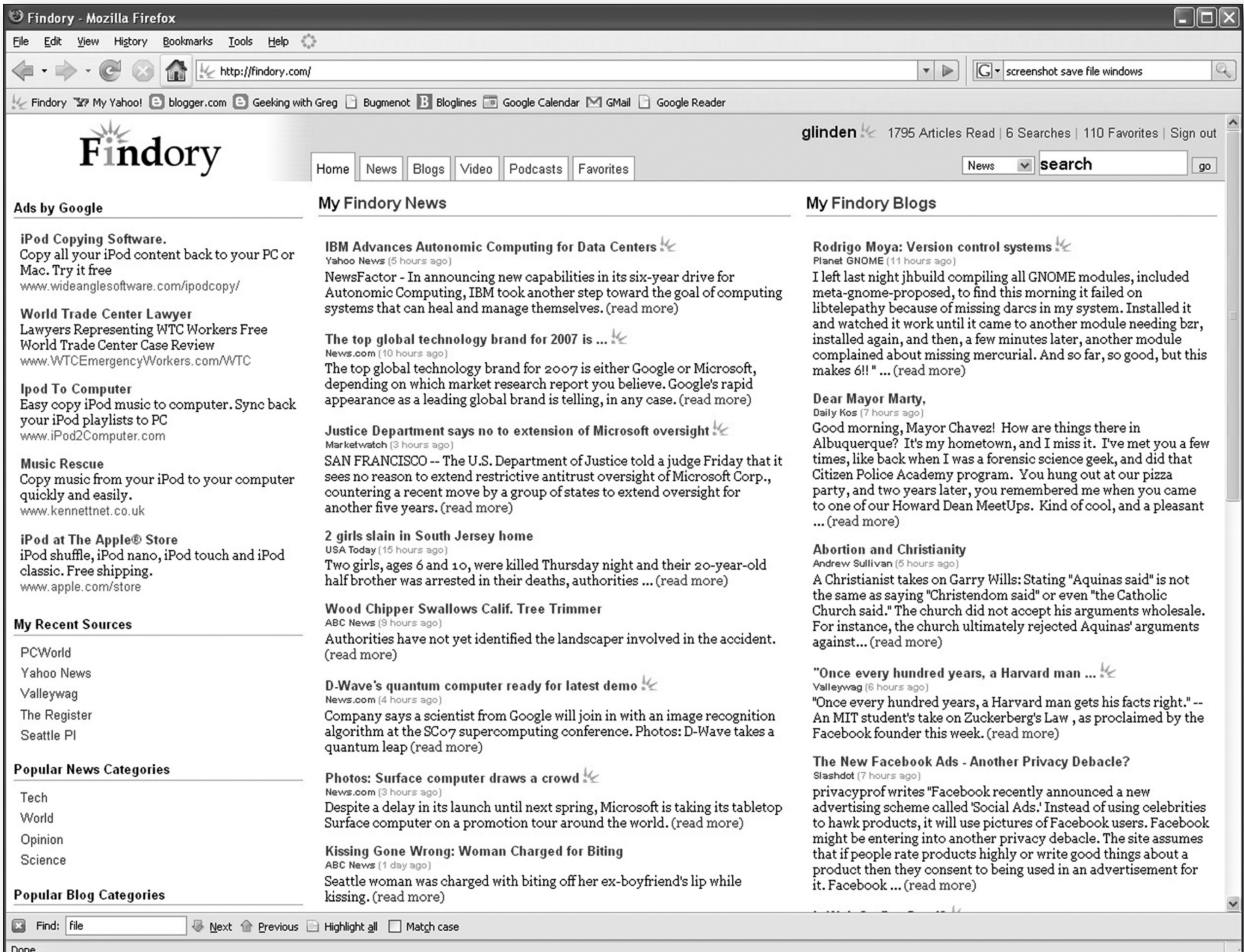

Figure 3. A Screenshot from Findory.

This figure is an example of a second-wave recommender system for news articles — a realization of Negroponte's (1995) "Daily Me" and a precursor of current personalized newspapers for the iPad. (Used with permission.)

for offering recommendations as an affordable software service (Clever Set, Loomia, Rich Relevance, Strands, and others) to smaller web companies. See table 1 for a more complete list. The common approach was for the vendor to offer a remote application programming interface (API) that collected user activity on web pages (click stream, ratings, purchases, and others), and generated on-the-fly recommendations relevant to the users' behavior. Despite strong evidence that different product and service domains (such as movies, music, news, and others) have different needs for recommender services, many software-as-service recommender system vendors generalized their solutions with only minor tailoring for specific domains. An additional problem for these vendors arose when web usage analytics companies started to incorporate the same recommender techniques as part of their service offerings.

Recommender systems have now become well studied both theoretically and practically. Table 3 shows a number of lessons learned deploying recommenders during the second wave. However, despite all the progress made, most commercial recommender systems have taken the business's side and their recommendations are firmly linked to short-term sales of products or services instead of considering real users' contexts and needs. Thus, we predict a new wave of recommender systems on the user's side that will better deal with information overload.

Figure 4. Tag Cloud Created from the Titles of All the Articles Published at the ACM RecSys Conferences Using IBM Wordle.

Hybridization Solves Many of the Issues with Recommender Systems and Wins Contests Too

Using exclusively one recommender systems technique produces less robust solutions. An ensemble of techniques has shown to be successful and easily improve accuracy.

Easy to Integrate Also Means Easy to Remove

Offering software-as-service recommender systems reduces the burden of integration and makes it easier to get the customers but also makes it easier for the customers to switch to another vendor.

Integration with Customer Care Services Is a Pending Subject

You can buy hundreds of products from Amazon or Apple and get better and better recommendations but unbelievably their customer care services do not know if you are a good or bad customer. Integration of CRM and recommender systems is a challenging issue.

Right Balance of Attention Needs to Be Given to Algorithms and User Experience

Although most research focuses on algorithms, user interfaces and user experience have a heavier weight on the final performance of recommender systems. Algorithms are mature, but an engineering approach to know how and when to recommend is still immature.

Cold Start Is Not a Problem If You Think Out of the Box

With the web 2.0 revolution, user-generated content made it possible to overcome the cold-start problem by just considering relatedness among items and people on the social web.

The Best Nearest Neighbor Algorithm Is the Social Graph

Do not spend time calculating nearest neighbors and just let people tell you!

Rating-systems-explicit rating systems are very often not well defined semantically and lead to poor recommendations. Implicit feedback on actual preferences have resulted in much better recommendations than explicit ratings.

Recommendation Technologies Have Been Commoditized

Technologies that were licensable for hundreds or even millions of dollars only 10 years ago are now freely available (See figure 2). This, together with the high number of players and the difficulties to evaluate or differentiate their products, has been one of the main causes that pushed many vendors out of business as the price for recommendation technologies dropped to levels that made operating their businesses unsustainable.

Table 3. Some Lessons Learned During the Second Wave of Recommender Systems.

The Third Wave of Recommender Systems

With more than 5 billion mobile subscribers, and more than 2 billion Internet users (see figure 2), we live in a era where information has gone from scarce to superabundant (Cukier 2010).

Technically savvy individuals are already utilizing a broad range of services, but must manage and prioritize the flows of information and suggestions on their own. On a typical day a given person may already be receiving a multitude of recommendations from mobile and nonmobile computing platforms. Personal sleep aid systems monitor sleep

Figure 5. A Screenshot from Siri.

This is an example of a third-wave recommender system and a precursor of the mobile personal assistant revolution.

cycles, making suggestions for healthy patterns of resting (such as Zeo, Fitbit). Personal workout recommenders may display jogging route recommendations (MapMyRun, Nike+, Strands, and others).

Personal news sites recommend articles from major news outlets or independent bloggers (such as Hacker News, Reddit, Digg, and others), while email systems help prioritize and filter our email (for example, Google gmail spam filtering and priority inbox). Personal assistant software schedules meetings or finds reservations for a restaurant (for example, Siri; see figure 5). Entertainment providers recommend movies and television programs to watch (such as Netflix).

Instead of providing yet another domain-specific recommendation service, we foresee a third wave of recommender systems that will return to the metaphor of *personal assistant* (Negroponte 1995, Maes 1994). In this arrangement, recommender systems once again persist with and act on behalf of individual users to help them cope with the problem of information overload across a range of domains. As the pace of data growth and consumption increases, the added dimensions of cloud and mobile computing enable better personalized recommendations that can be provided in a larger variety of relevant locations. Out of all of the recently started services, we see Siri (acquired by Apple) as coming closest to what the third-generation recommender systems will look like. In general, the third wave of recommender systems will have these prominent dimensions: contextual awareness, cloud-based stores and computation, and emphasis on users' long-term goals.

Contextual Awareness

New recommenders will use clues from mobile devices to generate *contextually appropriate* recommendations. Smartphones provide a more convenient and persistent basis for recommendations, as well as having access to new information, such as calendar, email, notifications, and user GPS coordinates, that helps drive timely and contextually appropriate suggestions. Additional technologies such as eye tracking, gesture detection, or skin-tension measurement are expected to play a role in the construction of such personal assistants, utilizing minimally invasive biometric feedback to characterize user affect or emotional state, according to relevant research by Rosalind Picard (2000). The advanced features and programmability of modern smartphones make them the perfect channel to drive the third generation of recommender systems.

Cloud-Based Stores and Computation

The amount and diversity of data that new recommender systems will process is going to increase dramatically while the cost to process or store will decrease significantly. During the first two waves, the IT investment required for large-scale recommendation systems was prohibitive for many vendors to pursue direct-to-consumer initiatives. However cloud computing has reshaped the landscape. On-demand computational resources, such as Amazon EC2 or Rackspace, combined with distributed-computing paradigms, such as map-reduce, provide supercomputing power at a fraction of its previous cost. At the same time, large varieties of information are being made available online through public APIs. In addition, new computing libraries, frameworks, and services such as SciPy, Apache Mahout, and Google Prediction API are making it easier for developers to utilize sophisticated algorithms on massive amounts of data. This will enable the third-wave recommender system ventures to focus more on the user experience and less on the technical side of algorithms.

User's Long-Term Goals

Finally, third-generation recommender systems will have a greater emphasis on handling long-term user goals and constraints, such as monthly budget limits, or scheduling constraints on the user's personal calendar. Most current recommendation systems focus on short-term interactions: cross-selling, up-selling, or suggesting media for immediate consumption. Removing the emphasis on short-term profit will allow recommendation systems to encourage discovery and, we hope, maximize a user's long-term satisfaction.

Conclusion

We characterize two basic types of recommender systems that have been deployed in practice. The first type includes recommenders that take the user's side (such as Siri) and are operated on the user's computer (smartphone or tablet) by a completely independent service provider. The second type corresponds to recommenders that take the business side (for example, Amazon's recommender). These recommenders are put in place and operated by the retailer of the product being recommended or operated on behalf of the retailer.

As technology advances and recommender systems become more commonplace for the public, we may now be on the verge of a third wave of recommender systems that considers the goals and activities of users in addition to their preferences and needs. The third wave of recommender systems will be capable of learning about a user's contextual behaviors and preferences to anticipate a user's actions at any point in time and act upon them.

It is an exciting time for recommender systems. The current confluence of ever faster networks, massive adoption of mobile devices, the rise of cloud computing and social technologies open a new window of opportunity. Companies like Facebook, Google, and Apple are well positioned given their dominance in social media, search, and mobile technology to capitalize on some of this opportunity. But the current industry scenario could not be better for both researchers and practitioners who want to address new challenges for recommender systems. There are ample opportunities for nimble entrepreneurs to connect data, algorithms, devices, and user needs to create solutions that carve out significant new market space.

References

Amatriain, X.; Torrens, M.; Resnick, P.; and Zanker, M., eds. 2010. *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. New York: Association for Computing Machinery.

Bergman, L. D.; Tuzhilin, A.; Burke, R. D.; Felfernig, A.; and Schmidt-Thieme, L., eds. 2009. *Proceedings of the Third ACM Conference on Recommender Systems (RecSys '09)*. New York: Association for Computing Machinery.

Cukier, K. 2010. Data, Data Everywhere: A Special Report on Managing Information. *The Economist*. 25 February.

Goldberg, D.; Nichols, D.; Oki, B. M.; and Terry, D. 1992. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM* 35(12): 61–70.

Jannach, D.; Zanker, M.; Felfernig, A.; and Friedrich, G. 2010. *Recommender Systems: An Introduction*. New York: Cambridge University Press.

Kirn, W. 2010. My Cart, My Self. *The New York Times Magazine*. 30 December.

Konstan, J. A.; Riedl, J.; and Smyth, B., eds. 2007. *Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys '07)*. New York: Association for Computing Machinery.

Kozierok, R., and Maes, P. 1993. A Learning Interface Agent for Scheduling Meetings. In *Proceedings of the 1993 International Workshop on Intelligent User Interfaces*. New York: Association for Computing Machinery.

Linden, G.; Smith, B.; and York, J. 2003. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing* 7(1): 76–80.

Maes, P. 1994. Agents That Reduce Work and Information Overload. *Communications of the ACM* 37(7): 30–40.

Negroponte, N. 1970. *The Architecture Machine: Toward a More Human Environment*. Cambridge, MA: The MIT Press.

Negroponte, N. 1995. *Being Digital*. New York: Vintage Books.

Palme, J. 1984. You Have 134 Unread Mail! Do You Want to Read Them Now? In *Proceedings of the IFIP WG 6.5 Working Conference on Computer-Based Document Sevices*, 175–184. Amsterdam, The Netherlands: Elsevier Noth-Holland, Inc.

Picard, R. 2000. *Affective Computing*. Cambridge, MA: The MIT Press.

Pu, P.; Bridge, D. G.; Mobasher, B.; and Ricci, F., eds. 2008. *Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys '08)*. New York: Association for Computing Machinery.

Resnick, P., and Varian, H. R. 1997. Recommender Systems. *Communications of the ACM* 40(3): 56–58.

Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; and Riedl, J. 1994. GroupLens: an Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, 175–186. New York: Association for Computing Machinery.

Ricci, F.; Rokach, L.; Shapira, B.; and Kantor, P. B., eds. 2011. *Recommender Systems Handbook*. Berlin: Springer.

Shardanand, U., and Maes, P. 1995. Social Information Filtering: Algorithms for Automating "Word of Mouth." In *Human Factors in Computing Systems CHI 95 Conference Proceedings*, 210–217. Reading, MA: Addison-Wesley.

Su, X., and Khoshgoftaa, T. M. 2009. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence* 2009: 1–20.

Töscher, A.; Jahrer, M.; and Bell, R. M. 2009. The BigChaos Solution to the Netflix Grand Prize. Technical Report, Commendo Research & Consulting GmbH, Köflach, Austria.

Francisco J. Martin is a scientist entrepreneur passionate about machine learning and distributed systems. He has cofounded three AI-related startups (Intelligent Software Components, SA, in 1999, Strands, Inc., in 2004, and BigML, Inc., in 2011) and directly raised more than $75 million in venture capital. He holds a five-year degree in computer science from Technical University of Valencia and received his Ph.D. (summa cum laude) from Technical University of Catalonia.

Justin Donaldson is cofounder and president of BigML, Inc., a web service for cloud-based machine learning of large sets of data. He has worked for six years on recommender systems at Strands.com, focusing on related analysis, interface, and visualization methods. He holds a B.S. in computer science from DePauw University, an M.S. in HCI, and a Ph.D. in informatics from Indiana University.

Adam Ashenfelter is a cofounder at BigML, Inc. He holds a B.S. and M.S. in computer science from Oregon State University. He spent seven years working on recommendation systems, machine learning, and data fusion projects while at CleverSet, Inc., and Strands, Inc.

Marc Torrens received a MSc degree in computer science from the Universitat Politècnica de Catalunya (UPC), Barcelona, Catalonia, in 1997, and a PhD from the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2002. He cofounded Iconomic Systems SA in 1999 and served the company as technology officer. Iconomic Systems developed software for planning travels on the Internet and was successfully sold to i:FAO AG, Germany. In 2004, Torrens cofounded Strands, Inc., in Oregon, USA. Strands develops recommendation and personalization technologies to help people discover new things. He is currently serving Strands as chief innovation officer. Torrens has published more than 20 referred papers on his work in international conferences and journals on artificial intelligence and usability.

Rick Hangartner is chief scientist at Strands Labs, Inc., and an adjunct faculty member at Oregon State University. He received his BA and MS from the University of South Florida and his PhD from Oregon State University. His research focuses on statistical inference, nonparametric data modeling, and adaptive systems.