Past, Present, and Future of Recommender Systems: An Industry Perspective

Xavier Amatriain
Quora
650 Castro St. #450
Mountain View, CA 94041, USA
xavier@quora.com

Justin Basilico
Netflix, Inc.
121 Albright Way
Los Gatos, CA 95032, USA
jbasilico@netflix.com

ABSTRACT

When the Netflix Prize launched in 2006, it put a spotlight on the importance and use of recommender systems in real-world applications. The competition provided many lessons, and many more have been learned since the Grand Prize was awarded in 2009. The use of recommender systems in industry has continued to grow driven by the availability of many kinds of user data and the continued interest for the area within the research community. In this paper, we will describe what we see as the past, present, and future of recommender systems from an industry perspective.

1. INTRODUCTION

Recommender Systems in industry are a main-stream application of large-scale machine learning, data-driven innovation, human computer interaction, and other research. Diverse application areas such as e-commerce, Internet music and video, news, search, gaming, and even online dating apply similar techniques that leverage large volumes of data to fulfill a user's needs in a personalized fashion. They have wide applicability because they can increase core business metrics such as customer satisfaction and revenue.

In this paper, we start by describing the main lessons from the Netflix Prize, which constitute, in our view, the past of recommender systems. We then describe the current landscape of recommender systems in industry and the main techniques currently used. Finally, we highlight some research directions that we think are potentially very valuable from an industry perspective.

2. PAST: THE NETFLIX PRIZE

In 2006, Netflix announced the Netflix Prize, a competition to predict movie ratings on a 5-star scale. Netflix conducted this competition to find new ways to improve its recommendations. However, they had to come up with a proxy question that was easier to evaluate and quantify: the root mean squared error (RMSE) between the predicted and actual rating. They offered a \$1 million prize to whomever provided a solution that reduced RMSE by 10% compared to their existing system, Cinematch. The Netflix Prize put a spotlight on the Recommender Systems area and highlighted the value of generating personalized recommendations. It was a landmark event that increased interest in the research area. Its crisp problem definition enabled thousands of teams to focus on improving a single metric. While this was a simplification of the recommendation problem, many valuable lessons were learned.

After the first year of competition, the KorBell team won the first Progress Prize with an 8.43% improvement. They reported more than 2000 hours of work to come up with a combination of 107 algorithms that put them at the top of the leaderboard. Netflix focused on the two best performing algorithms in the ensemble: a Matrix Factorization (MF) [12] called SVD++ and Restricted Boltzmann Machines (RBM) [22]. The original code was built to handle 100 million ratings and but Netflix had over 5 billion. Also, it was designed to run on a static dataset and not built to adapt as members added more ratings. Once Netflix overcame those challenges, they put both algorithms into production.

Many other learnings came out of the competition. For example, early on, it became clear that temporal dynamics was an important factor [11]. Another finding was that there is a large amount of noise in the ratings provided by users [2]. After almost three years, the final ensemble that won the grand prize was an impressive compilation of work, blending hundreds of models to finally cross the finish line [3]. It highlighted the power of ensembles to combine heterogeneous models and achieve maximum accuracy. Netflix evaluated some of the methods in the final solution, however, the additional increase in RMSE did not justify the large engineering effort to productionize them.

Recsys '16 September 15–19, 2016, Boston, MA, USA
© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-4035-9/16/09...\$15.00
DOI: http://dx.doi.org/10.1145/2959100.2959144

3. PRESENT: RECOMMENDATION BEYOND RATING PREDICTION

Many companies have discovered through the years that there is significant business value in incorporating recommendations to personalize much of their user experience. In this section we briefly survey different industrial scenarios for recommender systems as well as different formulations of the recommender problem that go beyond the paradigm of rating prediction.

3.1 Recommender Systems in Industry

Recommendation systems are used by many companies in a variety of application domains. Each domain has its own unique recommendation challenges. We provide here a short overview of some of the more well-known applications in industry.

Starting with Netflix, their approach to recommendations changed a lot since the Prize. Its service changed from DVD-rental-by-mail to instant video streaming across many devices. Netflix relies primarily on the recommendation system to help with people find something to watch that they will enjoy [7]. It does so by incorporating recommendations to personalize as much of the user experience as possible. Beyond Netflix, video recommendation has been an active area of research so, unsurprisingly, it is applied to recommend a variety of content spanning movies, TV shows, live, and user-generated. For instance, recommendations help people navigate the vast amounts of user-generated video within YouTube [6].

Music recommendation [4] is also an active application area. Pandora, for instance, centers their business model around creating personalized music stations. They combine collaborative filtering techniques with curated data called the Music Genome Project [24]. Spotify also provides personalized music recommendations in its service. EchoNest, a startup later acquired by Spotify, combined different approaches including collaborative filtering, metadata, and audio signal analysis [13].

Today most e-commerce sites and applications have a recommendation engine powering some of their user experience. The first large e-commerce company credited with a recommender system at the core of their experience is Amazon, who initially used a simple item-item collaborative filtering approach [14]. Their current experience includes many different levels of recommendation: from the homepage to many product pages having lists of other products bought or viewed. Other retail companies such as eBay have also incorporated recommendations in their experience [29].

News is another area where companies use recommendation approaches to personalize content for a user's interests. For example, Google News provided recommendations for news articles from the beginning [5]. For news recommendation, some of the key challenges are freshness, where relevant articles may have a very limited time span, and diversity, where there can be a large number of articles about the same topic. However, they have the advantage of using natural language processing techniques to create features.

Social networks also provided new recommendation scenarios. Twitter, for example, introduced their Who to Follow algorithm to recommend new social connections [9]. The combination of social with other kinds of signals [10] is a very common approach in products like Twitter or Facebook [16]. Quora has an overlay social network where different kinds of recommendations take place. Besides feeds and who to follow, Quora uses social signals to influence many kinds of recommendations. Another example of social recommendations are online dating sites, where the success of the system is not determined by one user receiving a good recommendation but rather by both parties accepting it [18]. Two-side recommendations are important for many social applications; Quora uses it to recommend which users might be able to answer a question.

Many of the previous examples illustrate the different or complimentary requirements that recommender systems have from an industrial point of view. Some important issues that are mentioned in most of these publications are scalability, business metrics, and integration of the system in the overall user experience. Typically the way algorithms are chosen in industry applications is by running randomized, controlled experiments (A/B tests) that measure the value of one approach over another.

3.2 Everything is a Recommendation

Personalization in many products starts on the homepage, which is usually displayed after login on a device. In the case of Netflix, this page consists of groups of videos arranged in horizontal rows that create a grid of videos scrollable in two dimensions. Each row has a title that conveys the intended connection between the videos in that row. Most of Netflix' recommendation approach is embodied in the way they generate rows, select rows, choose videos for a row, order the videos in a row, and order the rows on a page. On the other hand, Quora's homepage is organized around the central idea of a personalized feed, which is common in other social products like Facebook and Twitter, where recommended content items are displayed as a summary card and they can be scrolled vertically. These applications make recommendation front-and-center by delivering them as their primary user experience. With such an emphasis on the recommendations, an important element for all these systems is providing *awareness* and *explanations* for why an item was recommended.

While search or browsing is often provided as an alternative to using recommendations, even a situation where a user enters a search query can be an opportunity for recommendation. For example, the auto-complete suggestions when the user starts typing can be personalized and interpreted as recommendation over a constrained set. LinkedIn's Metaphor system is an example of a complete search recommendation system [20].

3.3 Beyond Explicit Feedback

Explicit ratings are neither the only feedback sites get from users nor the best kind of feedback. Implicit feedback is much more readily available than ratings and requires no extra user effort. For instance, on a web page you can have users visiting a URL or clicking on an link as a positive feedback. In a music service, a user can start playing a song. Also, many of these recommendation applications focus on helping a user choose an action (click, buy, read, listen, watch), so clearly information about previous actions is highly relevant for predicting future actions. That is why many recent recommendation approaches focus on the more reliable and readily available data from implicit feedback. For example, Bayesian Personalized Ranking (BPR) [21], uses it to compute a personalized ranking. Implicit and explicit feedback can also be combined in different ways, for example using SVD++, logistic ordinal regression [17] to provide a mapping, or taking a Bayesian approach like Matchbox [27].

3.4 Ranking

Central to many industrial recommenders is a personalized ranking system. Its goal is to find the best ordering of a set of items for a user within a specific context. Netflix and YouTube optimize ranking algorithms to put videos that a user is most likely to play and enjoy at the beginning of the list. Quora and Facebook do the same to optimize the probability that the user finds a story in Feed interesting. In all these cases, this is done by learning a scoring function $f_{\texttt{rank}} : U \times V \times C \to \mathbb{R}$ over users, items, and contexts.

An obvious baseline for a ranking function to optimize consumption is popularity. The reason is clear: on average, a user is most likely to be interested in what most others are interested in. However, popularity is the opposite of personalization; it will produce the same ordering of items for every user. Thus, the first goal of a personalized ranking function is to be better than item popularity and thus better satisfy users with varying tastes. One obvious way to approach this when rating data is available is to incorporate the predicted rating of each item. Rather than using either popularity or predicted rating on their own, we can learn to produce rankings that balance both of these aspects. This two-dimensional model is a basic example of a ranking function; real products use many other features.

The traditional *pointwise* approach for Learning to Rank (LTR) treats ranking as a simple binary classification problem where the only input are positive and negative examples. Typical models used in this context include Logistic Regression, or Gradient Boosted Decision Trees. *Pairwise* algorithms optimize a loss function defined on pairwise preferences to minimize the number of inversions. BPR [21] is a good example of such an approach. Going further, *listwise* approaches try to directly optimize the ranking of the whole list instead of individual pairs. RankALS [28], for example, defines an objective function that directly includes the ranking optimization and then uses Alternating Least Squares for optimizing. These approaches use rank-specific information retrieval metrics to measure the performance of a ranking model. Ideally, we would like to directly optimize our models those same metrics. However, they are not differentiable and standard methods cannot be directly applied. Some methods like CLiMF [25] use a smoothed version of the objective function to run gradient descent.

4. FUTURE: RESEARCH DIRECTIONS WITH INDUSTRIAL APPLICABILITY

In this section, we will describe some future research directions in Recommender Systems for industry use. It is important to note that there are other issues, such as user and item cold-starting or finding the right metric that correlates to long-term user satisfaction, that are far from being solved. These issues, however, have already been discussed at length in the research literature. Our goal here is to focus the attention of the research community in directions that are very important for industry but are less well studied.

4.1 Full Page Optimization

While one-dimensional ranking is already a step beyond rating prediction and can be further optimized for presentation [23], we are most interested in optimizing a recommendation experience over a complete page. Generating a full page that is personalized and optimized is complex. In the Netflix scenario, the set of available candidate rows is much larger than the catalog of individual items since the same item can be included in many different rows and the row candidates need to be selected and ranked together. Also, when optimizing a whole page, we not only optimize for relevance, but also other aspects like diversity and freshness. An ideal full page optimization algorithm would also be able to personalize how recommendations are mixed with other elements of the user experience by understanding a user's browsing or attention behavior [15]. It may also be useful to adapt a page within a session as a user's actions reveal their current intents.

While this is not yet a common theme in the literature, there are a few recent papers that address the page optimization area. For example, [1] present an approach in the context of news involving a sequential click model for the user and a relevance model that promotes diversity through submodular functions. Another example is [30], where the layout of elements on a search results page is performed using a learned function over both the page elements and layout.

4.2 Personalizing How We Recommend

Another direction for improvement in recommendation systems is personalizing not just *what* we recommend but also *how* we recommend. For example, many recommendation approaches have assumptions or parameters balancing how much diversity, novelty, popularity, etc. the system to produces. However, different people can vary quite a bit in their desired levels of these aspects. When doing recommendations internationally, as is needed in many industrial applications, this also means properly adapting to the balance to needs and tastes of diverse cultures and languages. Another area is personalizing how recommendations are displayed. For a given item, there are many options on how it could be presented, along with many pieces of supporting evidence to help a user decide whether or not to choose it. There is the opportunity to make recommendations even more powerful by surfacing the most useful information for that specific user to make a decision. This goes beyond just generating explanations from recommendation algorithm itself but instead considering all of the aspects of presentation (images, descriptions, metadata, connections to other items, etc.) as elements that can be personalized. In this Netflix case, this may mean deciding whether or not to show explanations, reviews, average ratings, play trailers, cast information, and so forth, based on which ones are most likely to result in high-quality choices. Going further, the entire interaction paradigm for a recommendation system could be personalized.

4.3 Indirect Feedback

While we have seen a shift in industry from explicit to implicit feedback, most work has focused on direct forms of feedback: a user directly performing an action on an item. A shortcoming in using standard implicit feedback is that we do not see negative feedback; the data is either positive or missing. The missing data includes both items that the user explicitly chose to ignore because they were not appealing and items that would have been perfect recommendations but were never presented to the user [26]. One piece of information that can help is knowing if items were displayed to the user. This adds very valuable information, but slightly complicates the formulation of our recommendation problem. Approaches such as Collaborative Competitive Filtering (CCF) [31] can use this information by modeling not only the collaboration between similar users and items, but also the competition between items for user attention. A related issue related to how items are presented is the so-called position bias: An item that is presented in the first position of a list is more likely to be seen and chosen than one that is further down [19]. Thus, a deeper understanding of causality has a lot of potential to help recommendation systems.

As someone interacts with an application that involves recommendations, a lot of other, indirect forms of information about how they navigate is also collected. Also, indirect feedback is linked with a lot of contextual information. In the case of Netflix, for example, the user's preference for shows might depend on variables such as time of the day, day of the week, or viewing device. There is the potential for effective improvements in applications that make use of context [8]. However, indirect and contextual variables represents having to deal with more data that is often noisy and creates a higher dimensional problem space.

4.4 Value-aware Recommendations

The recommender systems community has traditionally separated recommender systems into two kinds: content-based and collaborative filtering. Nowadays, both techniques are combined in many different ways. For example, content-based recommendations are used as a solution for cold-start. However, in both cases, data extracted from the content or the user interactions is used to optimize the same thing: the probability a user uses the recommendation. Unfortunately, this formulation is based on an assumption that all content has the same value. However, in many applications, this is not the case.

In e-commerce some items might have a higher margin for the retailer while giving the same satisfaction to the user [32]. In media/news recommendation, we might have content that is very similar on the surface but are vastly different levels of quality. For example, Quora is interested in recommending items of a certain quality so that the norms of the social network evolve in a way where quality is valued. Most online services will care about recommending the kind of content that drives long-term retention as opposed to the kind of content that might increase short-term clicks but might not result in long-term user satisfaction. We also may want to make sure that our algorithms are not biased against certain kinds of users, for example not reinforcing prejudices. As such, it is important to also keep in mind a user's values.

Many products define additional rules to filter or promote content that thought to have a positive long-term effect in one of the dimensions mentioned above. However, there is a clear need to come up with algorithms that care holistically about the long-term value of the item being recommended so that given a similar probability user engagement, the algorithm can steer towards long-term gains. We call this family of algorithms *value-aware recommendations*.

An extreme and forward-looking version of this would be for recommendation systems to additionally recommend what new items or content that does not exist should be created to fill missing needs and add the most value. Companies are already doing this to some extent in evaluating items. Netflix uses data to decide whether or not a piece of content should be licensed. Quora uses the probability of authors creating good quality and engaging answers as a way to route questions to the right people. However, there is a lot of potential impact in researching how to make recommendations for what new items would add the most value.

5. CONCLUSION

The Netflix Prize abstracted the recommendation problem to a simplified proxy of accurately predicting ratings. It is now clear that this is just one of many components in an effective industrial recommendation system. They also need to account for factors like diversity, context, evidence, freshness, and novelty. Balancing these often competing factors can be a daunting task but we have found that it is best handled using a range of algorithmic approaches and many types of data. We also need optimized approaches, appropriate metrics, rapid experimentation frameworks, solid algorithmic techniques, great user interfaces, and scalable architectures embedded within a sound methodology for figuring out what actually improves the user experience. All of this is driven forward by the innovation happening in the Recommendation System community. When industry and research work hand-in-hand to define the future, we find ourselves continually making progress towards that goal of creating the best possible experience for people all over the world.

6. REFERENCES

[1] A. Ahmed, C. Teo, S.V.N. Vishwanathan, and A. Smola. Fair and balanced: Learning to present news stories. WSDM '12.

[2] X. Amatriain, J. M. Pujol, and N. Oliver. I Like It... I Like It Not: Evaluating User Ratings Noise in Recommender Systems. In *UMAP '09*. 2009.

[3] R. M. Bell and Y. Koren. Lessons from the Netflix Prize Challenge. *SIGKDD Explor. Newsl.*, 9(2), December 2007.

[4] O. Celma. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play*. 2010.

[5] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: Scalable online collaborative filtering. WWW '07, 2007.

[6] J. Davidson et. al. The youtube video recommendation system. RecSys '10, 2010.

[7] C. Gomez-Uribe and N. Hunt. The Netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4), December 2015.

[8] M. Gorgoglione, U. Panniello, and A. Tuzhilin. The effect of context-aware recommendations on customer purchasing behavior and trust. RecSys '11, 2011.

[9] P. Gupta, Goel, Lin, Sharma, Wang, and Zadeh. WTF: The Who to Follow Service at Twitter. WWW '13.

[10] M. Jamali and M. Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. KDD '09, 2009.

[11] Y. Koren. Collaborative filtering with temporal dynamics. KDD '09, 2009.

[12] Y. Koren, R. Bell, and C. Volinsky. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8), August 2009.

[13] P. Lamere. I've got 10 million songs in my pocket: Now what? RecSys '12, 2012.

[14] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), January 2003.

[15] V. Navalpakkam, L. Jentzsch, R. Sayres, S. Ravi, A. Ahmed, and A. Smola. Measurement and modeling of eye-mouse behavior in presence of nonlinear page layouts. WWW '13.

[16] W. Oremus. Who controls your Facebook feed. *Slate*, January 2016.

[17] D. Parra, A. Karatzoglou, X. Amatriain, and I. Yavuz. Implicit feedback recommendation via implicit-to-explicit ordinal logistic regression mapping. In *Proc. of the 2011 CARS Workshop*, 2011.

[18] L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay. Recon: A reciprocal recommender for online dating. RecSys '10, 2010.

[19] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? CIKM '08, 2008.

[20] A. Reda, Y. Park, M. Tiwari, C. Posse, and S. Shah. Metaphor: A system for related search recommendations. CIKM '12, 2012.

[21] S. Rendle, C. Freudenthaler, Z. Gantner, and L. S. Thieme. BPR: Bayesian personalized ranking from implicit feedback. UAI '09, 2009.

[22] R. Salakhutdinov, Mnih, and Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proc of ICML '07*.

[23] O. Sar Shalom, N. Koenigstein, U. Paquet, and H. Vanchinathan. Beyond collaborative filtering: The list recommendation problem. WWW '16, 2016.

[24] Science. Rockin' to the Music Genome. *Science*, 311(5765), 2006.

[25] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic. CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering. RecSys '12.

[26] H. Steck. Training and testing of recommender systems on data missing not at random. In *Proc. of KDD '10*, 2010.

[27] D. H. Stern, R. Herbrich, and T. Graepel. Matchbox: large scale online bayesian recommendations. WWW '09, 2009.

[28] G. Takács and D. Tikk. Alternating least squares for personalized ranking. RecSys '12, 2012.

[29] J. Wang, B. Sarwar, and N. Sundaresan. Utilizing related products for post-purchase recommendation in e-commerce. RecSys '11, 2011.

[30] Y. Wang, D. Yin, L. Jie, P. Wang, M. Yamada, Y. Chang, and Q. Mei. Beyond ranking: Optimizing whole-page presentation. WSDM '16, 2016.

[31] S.H. Yang, B. Long, A.J. Smola, H. Zha, and Z. Zheng. Collaborative competitive filtering: learning recommender using context of user choice. SIGIR '11, 2011.

[32] Y. Zhang, Q. Zhao, Y. Zhang, D. Friedman, M Zhang, Y. Liu, and S. Ma. Economic recommendation with surplus maximization. In *WWW '16*, 2016.