Beyond Data: From User Information to Business Value through Personalized Recommendations and Consumer Science

Xavier Amatriain
Netflix
Los Gatos, CA 95032, USA
xavier@netflix.com

ABSTRACT

Since the Netflix $1 million Prize, announced in 2006, Netflix has been known for having personalization at the core of our product. Our current product offering is nowadays focused around instant video streaming, and our data is now many orders of magnitude larger. Not only do we have many more users in many more countries, but we also receive many more streams of data. Besides the ratings, we now also use information such as what our members play, browse, or search.

In this paper I will discuss the different approaches we follow to deal with these large streams of user data in order to extract information for personalizing our service. I will describe some of the machine learning models used, and their application in the service. I will also describe our data-driven approach to innovation that combines rapid offline explorations as well as online A/B testing. This approach enables us to convert user information into real and measurable business value.

Categories and Subject Descriptors

H.3.5 [**Information Systems**]: Information Search and Retrieval—*Online Information Services*

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

http://dx.doi.org/10.1145/2505515.2514701.

Keywords

Recommender Systems, Personalization, Machine Learning, Big Data

1. INTRODUCTION

Recommender Systems (RS) are a prime example of the mainstream applicability of large scale data mining. These systems leverage user data in order to produce a personalized experience that allows user to navigate large collection by zooming into their particular taste. For services such as Netflix, RS are at the core of their offering, and the value of the systems is directly linked to the business success.

There is more to a good recommender system than the data mining technique. Issues such as the user interaction design, outside the scope of this paper, may have a deep impact on the effectiveness of an approach. But given an existing application, an improvement in the algorithm can have a value of millions of dollars, and can even be the factor that determines the success or failure of a business. On the other hand, given an existing method or algorithm, adding more features coming from different data sources can also result in a significant improvement. I will describe the use of data, models, and other personalization techniques at Netflix in section 4.

Another important issue is how to measure the success of a given personalization technique. Root mean squared error (RMSE) was the offline evaluation metric of choice in the Netflix Prize (see Section 2). But there are many other relevant metrics that, if optimized, would lead to different solutions - think, for example, of ranking metrics such as Normalized Discounted Cumulative Gain (NDCG) or other information retrieval ones such as recall or area under the curve (AUC). Beyond the optimization of a given offline metric, what we are really pursuing is the impact of a method on the business. Is there a way to relate the goodness of an algorithm to more customer-facing metrics such as click-through rate (CTR) or retention? I will describe our approach to innovation called "Consumer Data Science" in section 3.

But before we understand the reasons for all these effects, let us take a step back and take a look at the Netflix Prize, and some of the lessons we learned.

2. THE NETFLIX PRIZE

In 2006, Netflix announced the Netflix Prize, a machine learning and data mining competition for movie rating prediction. We offered $1 million to whoever improved the accuracy of our existing system called Cinematch by 10%. We conducted this competition to find new ways to improve the recommendations we provide to our members, which is a key part of our business. However, we had to come up with a proxy question that was easier to evaluate and quantify: the root mean squared error (RMSE) of the predicted rating.

The Netflix Prize put the spotlight on Recommender Systems and the value of user data to generate personalized recommendations. It did so by providing a crisp problem definition that enabled thousands of teams to focus on improving a metric. While this was a simplification of the recommendation problem, there were many lessons learned.

2.1 Lessons from the Prize

A year into the competition, the Korbell team won the first Progress Prize with an 8.43% improvement. They reported more than 2000 hours of work in order to come up with the final combination of 107 algorithms that gave them this prize. And they gave us the source code. We looked at the two underlying algorithms with the best performance in the ensemble: Matrix Factorization (MF) [1] and Restricted Boltzmann Machines (RBM). Matrix Factorization by itself provided a 0.8914 RMSE, while RBM alone provided a competitive but slightly worse 0.8990 RMSE. A linear blend of these two reduced the error to 0.88. To put these algorithms to use, we had to work to overcome some limitations, for instance that they were built to handle 100 million ratings, instead of the more than 5 billion that we have, and that they were not built to adapt as members added more ratings. But once we overcame those challenges, we put the two algorithms into production, where they are still used as part of our recommendation engine.

The standard matrix factorization decomposition provides user factor vectors $U_u \in R^f$ and item-factors vector $V_v \in R^f$. In order to predict a rating, we first estimate a baseline $b_{uv} = \mu + b_u + b_v$ as the user and item deviation from average. The prediction can then be obtained by adding the product of user and item factors to the baseline as $r'_{uv} = b_{uv} + U_u^T U_v$.

One of the most interesting findings during the Netflix Prize came out of a blog post. Simon Funk introduced an incremental, iterative, and approximate way to compute the SVD using gradient descent [4]. This provided a practical way to scale matrix factorization methods to large datasets.

Another enhancement to matrix factorization methods was Koren *et. al*'s SVD++ [7]. This asymmetric variation enables adding both implicit and explicit feedback, and removes the need for parameterizing the users.

The second model that proved successful in the Netflix Prize was the Restricted Boltzmann Machine (RBM). RBM's can be understood as the fourth generation of Artificial Neural Networks - the first being the Perceptron popularized in the 60s; the second being the backpropagation algorithm in the 80s; and the third being Belief Networks (BNs) from the 90s. RBMs are BNs that restrict the connectivity to make learning easier. RBMs can be stacked to form Deep Belief Nets (DBN). For the Netflix Prize, Salakhutditnov *et al.* proposed an RBM structure with binary hidden units and softmax visible units with 5 biases only for the movies the user rated [9].

Many other learnings came out of the Prize. For example, early in the prize, it became clear that it was important to take into account temporal dynamics in the user feedback [8]. Another finding of the Netflix Prize was the realization that user explicit ratings are noisy. This was already known in the literature. Herlocker *et al.*[5] coined the term "magic barrier" to refer to the limit in accuracy in a recommender system due to the natural variability in the ratings. This limit was in fact relatively close to the actual Prize threshold [2], and might have played a role in why it took so much effort to squeeze the last fractions of RMSE.

The final Grand Prize ensemble that won the $1M two years later was a truly impressive compilation and culmination of years of work, blending hundreds of predictive models to finally cross the finish line [3]. The way that the final solution was accomplished by combining many independent models also highlighted the power of using ensembles.

At Netflix, we evaluated some of the new methods included in the final solution. The additional accuracy gains that we measured did not seem to justify the engineering effort needed to bring them into a production environment. Also, our focus on improving Netflix personalization had by then shifted from pure rating prediction to the next level.

3. CONSUMER DATA SCIENCE

Netflix has discovered through the years that there is tremendous value in incorporating recommendations to personalize as much of the experience as possible. This realization pushed us to propose the Netflix Prize described in the previous section. In the following sections, we will describe the main components of Netflix personalization. But first let us take a look at how we manage innovation in this space.

The abundance of source data, measurements and associated experiments allow Netflix not only to improve our personalization algorithms but also to operate as a data-driven organization. We have embedded this approach into our culture since the company was founded, and we have come to call it Consumer (Data) Science. Broadly speaking, the main goal of our Consumer Science approach is to innovate for members effectively. We strive for an innovation that allows us to evaluate ideas rapidly, inexpensively, and objectively. And once we test something, we want to understand why it failed or succeeded. This lets us focus on the central goal of improving our service for our members.

So, how does this work in practice? It is a slight variation on the traditional scientific process that iterates over the following steps:

1. **Start with a hypothesis**: Algorithm/feature/design X will increase member engagement with our service and ultimately member retention.

2. **Design a test**: Develop a solution or prototype. Think about issues such as dependent & independent variables, control, and significance.

[1]The application of Matrix Factorization to the task of rating prediction closely resembles the technique known as Singular Value Decomposition used, for example, to identify latent factors in Information Retrieval. Therefore, it is common to see people referring to this MF solution as SVD.

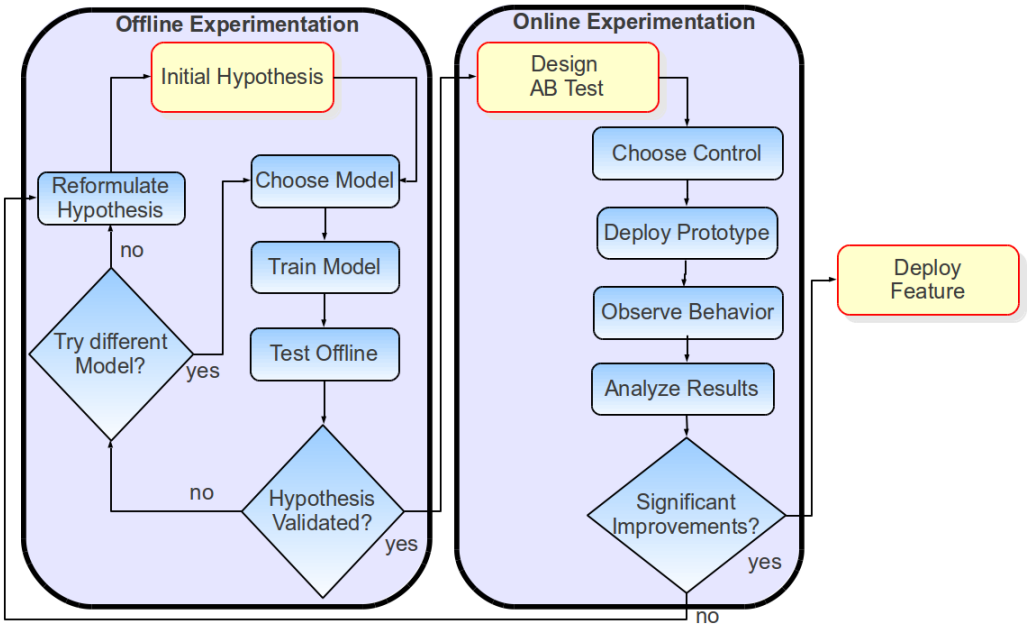

Figure 1: Following an iterative and data-driven offline-online process for innovating in personalization

3. **Execute the test**: Assign users to the different buckets and let them respond to the different experiences.

4. **Let data speak for itself**: Analyze significant changes on primary metrics and try to explain them through variations in the secondary metrics. If the hypothesis is validated, deploy, else reformulate the hypothesis and start over.

The tests that are executed during this process are the so-called A/B tests (or bucket tests), where comparable subsets of our population are exposed to different experiences in order to analyze their different responses. A traditional A/B tests will have only two experiences (A and B). However, typical A/B tests at Netflix will have between 5 and 20 cells, exploring variations of a base idea. Tests usually have thousands of members, and we typically have scores of A/B tests running in parallel. A/B tests let us try radical ideas or test many approaches at the same time, but the key advantage is that they allow our decisions to be data-driven.

When we execute A/B tests, we track many different metrics. But we ultimately trust member engagement (e.g. viewing hours) and retention. Retention is our Overall Evaluation Criteria (OEC) [6]. It measures the percentage of users who decide to stay with the service, and therefore pay the next monthly fee. It is a very crisp and valuable metric since it maps directly to our business success. Unfortunately, it is a slow metric that requires several months for a good analysis. That is why we also use metrics such as the hours streamed. Those metrics also relate to user engagement, but they are more sensitive, and response quicker to changes in the service.

An interesting follow-up question that we have faced is how to integrate our machine learning algorithmic approaches into this data-driven A/B test culture at Netflix.

To measure model performance offline we track multiple metrics: from ranking measures such as normalized discounted cumulative gain, to classification metrics such as precision, and recall. We also use the famous RMSE from the Netflix Prize or other more exotic metrics to track different aspects like diversity. We keep track of how well those metrics correlate to measurable online gains in our A/B tests. However, since the mapping is not perfect, offline performance is used only as an indication to make informed decisions on follow up tests.

The advantage of offline testing is that we can test many hypothesis - or variations of a hypothesis - inexpensively, and in little time. Instead of running costly and long A/B tests, we can execute many parallel offline experiments using existing data.

Once offline experimentation has validated a hypothesis, we are ready to design and launch the A/B test that will prove the new feature valid from a member perspective. The integration of offline algorithmic experimentation with online A/B testing defines an offline-online testing process that combines the best of both worlds (see Figure 1). The original iterative process presented above now becomes:

1. **Start with a hypothesis**: Algorithm/feature/design X will increase member engagement with our service and ultimately member retention.

2. **Design an offline experiment**: Using existing datasets train models and decide what metrics to optimize.

3. **Execute the experiment**: Evaluate the models with existing data and evaluate on the chosen metric(s)

4. **Evaluate experimental results**: If offline metrics are improved in a significant way, proceed with online evaluation. Else, reformulate hypothesis, and start over.

5. **Design a test**: Develop a solution or prototype. Think about issues such as dependent & independent variables, control, and significance.
6. **Execute the test**: Assign users to the different buckets and let them respond to the different experiences.
7. **Let data speak for itself**: Analyze significant changes on primary metrics and try to explain them through variations in the secondary metrics. If the hypothesis is validated, deploy, else reformulate the hypothesis and start over.

If the final A/B test shows positive results, we will be ready to roll out in our continuous pursuit of the better product for our members. That is in fact how we came about to having the personalization experience I will describe in the next section.

4. NETFLIX PERSONALIZATION: EVERYTHING IS A RECOMMENDATION

Personalization starts on our homepage in any device. This page consists of groups of videos arranged in horizontal rows. Each row has a title that conveys the intended meaningful connection between the videos in that group. Most of our personalization is based on the way we select rows, how we determine what items to include in them, and in what order to place those items.

Take as a first example the Top 10 row (see Figure 2). This row is our best guess at the ten titles you are most likely to enjoy. Of course, when we say "you", we really mean everyone in your household. It is important to keep in mind that NetflixâĂŹ personalization is intended to handle a household that is likely to have different people with different tastes. That is why when you see your Top 10, you are likely to discover items for dad, mom, the kids, or the whole family. Even for a single person household we want to appeal to your range of interests and moods. To achieve this, in many parts of our system we are not only optimizing for **accuracy**, but also for **diversity**.

Another important element in Netflix' personalization is **awareness**. We want members to be aware of how we are adapting to their tastes. This not only promotes trust in the system, but encourages members to give feedback that will result in better recommendations. A different way of promoting trust with the personalization component is to provide **explanations** as to why we decide to recommend a given movie or show (see Figure 3). We are not recommending it because it suits our business needs, but because it matches the information we have from you: your explicit taste preferences and ratings, your viewing history, or even your friends' recommendations.

On the topic of friends, we recently released our Facebook connect feature. Knowing about your friends not only gives us another signal to use in our personalization algorithms, but it also allows for different rows that rely mostly on your social circle to generate recommendations.

Some of the most recognizable personalization in our service is the collection of "genre" rows. These range from familiar high-level categories like "Comedies" and "Dramas" to highly tailored slices such as "Imaginative Time Travel Movies from the 1980s". Each row represents 3 layers of personalization: the choice of genre itself, the subset of titles selected within that genre, and the ranking of those

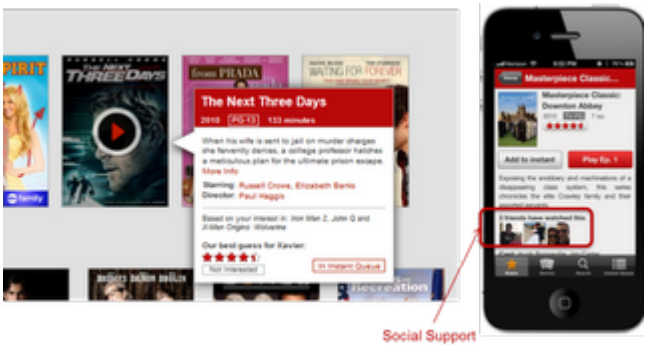

Figure 3: Adding explanation and support for recommendations contributes to user satisfaction and requires specific algorithms. Support in Netflix can include your predicted rating, related shows you have watched, or even friends who have interacted with the title.

titles. Rows are generated using a member's implicit genre preferences – recent plays, ratings, and other interactions –, or explicit feedback provided through our taste preferences survey (see Figure 4) . As with other personalization elements, **freshness** and diversity is taken into account when deciding what genres to show from the thousands possible.

Similarity is also an important source of personalization. We think of similarity in a very broad sense; it can be between movies or between members, and can be in multiple dimensions such as metadata, ratings, or viewing data. Furthermore, these similarities can be blended and used as features in other models. Similarity is used in multiple contexts, for example in response to generate rows of "adhoc genres" based on similarity to titles that a member has interacted with recently.

In most of the previous contexts, the goal of the recommender systems is still to present a number of attractive items for a person to choose from. This is usually accomplished by selecting some items and sorting them in the order of expected enjoyment (or *utility*). Since the most common way of presenting recommended items is in some form of list, we need an appropriate **ranking** model that can use a wide variety of information to come up with an optimal sorting of the items. In the next section, we will go into some of the details of how to design such a ranking model.

4.1 Ranking

The goal of a ranking system is to find the best possible ordering of a set of items for a user, within a specific context, in real-time. We optimize ranking algorithms to give the highest scores to titles that a member is most likely to play and enjoy.

If you are looking for a ranking function that optimizes consumption, an obvious baseline is item popularity. The reason is clear: on average, a member is most likely to watch what most others are watching. However, popularity is the opposite of personalization: it will produce the same ordering of items for every member. Thus, the goal becomes to find a personalized ranking function that is better than item popularity, so we can better satisfy members with varying tastes.

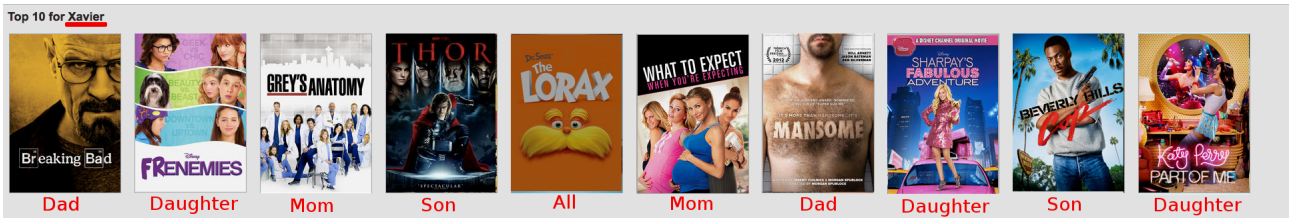

Figure 2: Example of a Netflix Top 10 row. We promote personalization awareness and reflect on the diversity of a household. Note though that personal labels are only the author's guess since the system is uncertain about the true household composition.

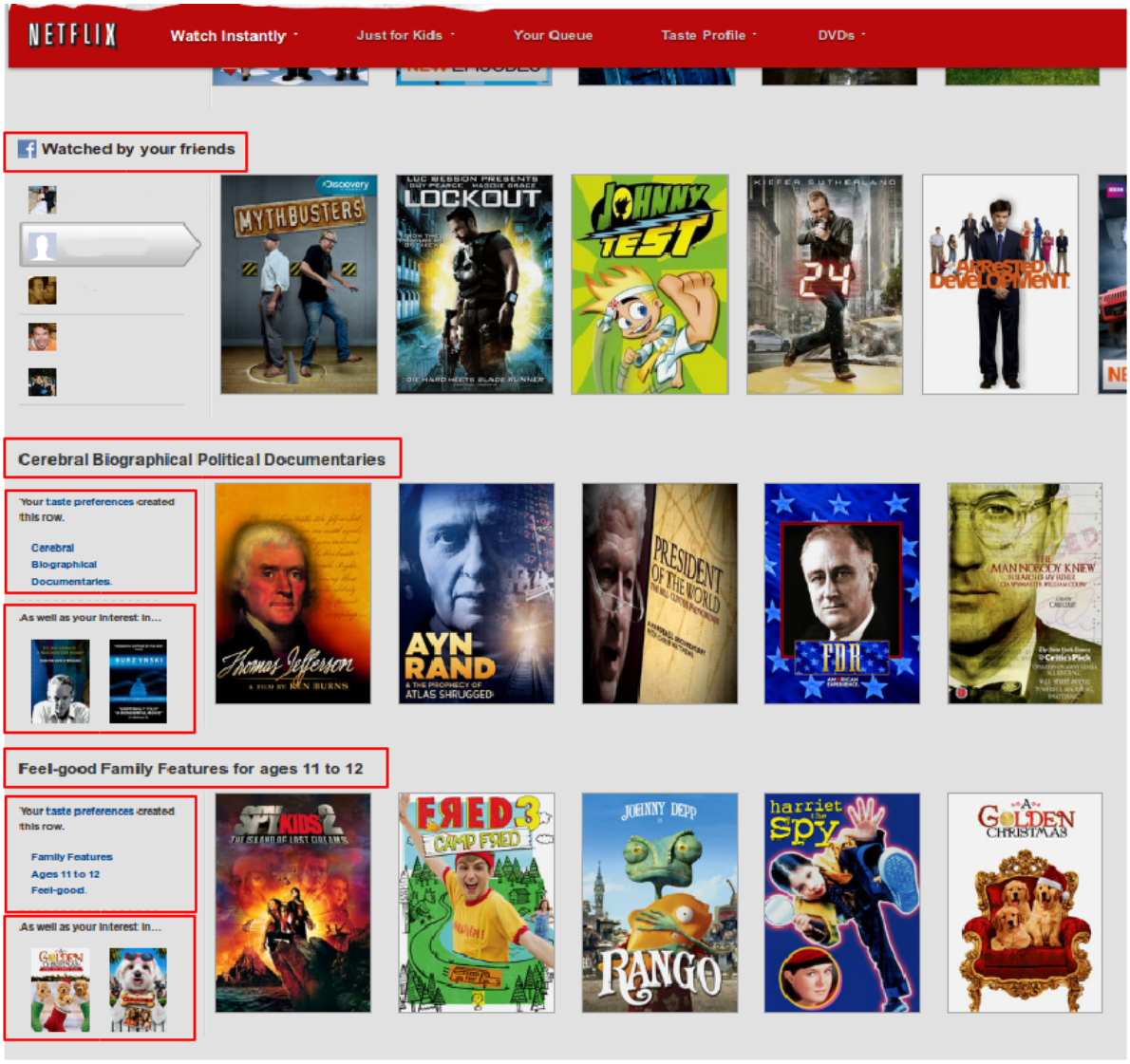

Figure 4: Netflix Genre rows can be generated from implicit, explicit, or hybrid feedback

Recall that our goal is to recommend the titles that each member is most likely to play and enjoy. One obvious way to approach this is to use the member's predicted rating of each item as an adjunct to item popularity. Using predicted ratings on their own as a ranking function can lead to items that are too niche or unfamiliar, and can exclude items that the member would want to watch even though they may not rate them highly. To compensate for this, rather than using either popularity or predicted rating on their own, we would like to produce rankings that balance both of these aspects. At this point, we are ready to build a ranking prediction model using these two features.

Let us start with a very simple scoring approach by choosing our ranking function to be a linear combination of popularity and predicted rating. This gives an equation of the form $score(u, v) = w_1 p(v) + w_2 r(u, v) + b$, where u=user, v=video item, p=popularity and r=predicted rating. This equation defines a two-dimensional space.

Once we have such a function, we can pass a set of videos through our function and sort them in descending order according to the score. First, though, we need to determine the weights w_1 and w_2 in our model (the bias b is constant and thus ends up not affecting the final ordering). We can formulate this as a machine learning problem: select positive and negative examples from your historical data and let a machine learning algorithm learn the weights that optimize our goal. This family of machine learning problems is known as "Learning to Rank" and is central to application scenarios such as search engines or ad targeting. A crucial difference in the case of ranked recommendations is the importance of personalization: we do not expect a global notion of relevance, but rather look for ways of optimizing a personalized model.

As you might guess, the previous two-dimensional model is a very basic baseline. Apart from popularity and rating prediction, we have tried many other features at Netflix. Some have shown no positive effect while others have improved our ranking accuracy tremendously. Figure 5 shows the ranking improvement we have obtained by adding different features and optimizing the machine learning algorithm.

Many methods can be used for personalized ranking: from traditional supervised classification approaches to advanced list-wise learning to rank models that directly optimize ranking metrics (see [1] for many pointers to interesting research on this area).

4.2 Data

The previous discussion on the ranking algorithms highlights the importance of both data and models in creating an optimal personalized experience. The availability of high volumes of high quality user data allows for some approaches that would have been unthinkable just a few years back. As an example, here are some of the data sources we can use at Netflix to optimize our recommendations:

- We have several billion item **ratings** from members. And we receive millions of new ratings every day.
- We already mentioned the use of global item **popularity** for ranking. There are many ways to compute popularity such as over various time ranges or grouping members by region or other similarity metrics.
- Our members add millions of items to their **queues** each day. And they directly enter millions of **search terms** each day.
- Each item in our catalog has rich **metadata** such as actors, director, genre, parental rating, or reviews.
- Using presentation and **impression data**, we know what items we have recommended and where we have shown them, and can look at how that decision has affected the user's actions. We can also observe the member's interactions with the recommendations: scrolls, mouse-overs, clicks, or the time spent on a given page.
- **Social** data has become our latest source of personalization features. Social data may include the social network connections themselves as well as interactions, or activities of connected nodes.
- We can also tap into **external data** such as box office performance or critic reviews to improve our features.
- And that is not all: there are many other features such as **demographics**, **location**, **language**, or **temporal data** that can be used in our predictive models.

4.3 Models

So, what about the models? Many different modeling approaches have been used for building personalization engines. One thing we have found at Netflix is that with the great availability of data, both in quantity and types, a thoughtful approach is required to model selection, training, and testing. We use all sorts of machine learning approaches: From unsupervised methods such as **clustering** algorithms to a number of supervised classifiers that have shown optimal results in various contexts. This is an incomplete list of methods you should probably know about if you are working in machine learning for personalization: **Linear regression**, **Logistic regression**, **Elastic nets**, **Singular Value Decomposition**, **Restricted Boltzmann Machines**, **Markov Chains**, **Latent Dirichlet Allocation**, **Association Rules**, **Matrix factorization**, **Gradient Boosted Decision Trees**, **Random Forests**, and Clustering techniques from the simple **k-means** to graphical approaches such as **Affinity Propagation**.

There is no easy answer to how to choose which model will perform best in a given problem. The simpler your feature space is, the simpler your model can be. But it is easy to get trapped in a situation where a new feature does not show value because the model cannot learn it. Or, the other way around, to conclude that a more powerful model is not useful simply because you don't have the feature space that exploits its benefits.

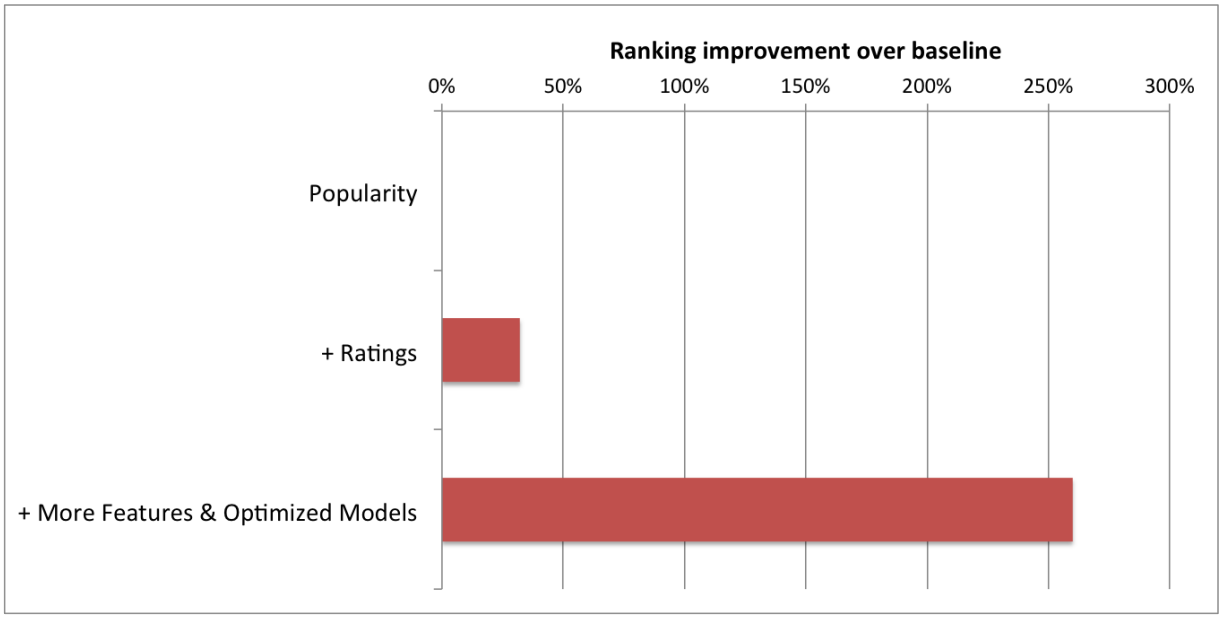

Figure 5: Performance of Netflix ranking system when adding features

5. CONCLUSIONS

The Netflix Prize abstracted the recommendation problem to a proxy and simplified question of predicting ratings. But it is clear that the Netflix Prize objective, accurate prediction of a movie's rating, is just one of the many components of an effective recommendation system. We also need to take into account factors such as context, popularity, interest, evidence, novelty, diversity, or freshness. Supporting all the different contexts in which we want to make recommendations requires a range of algorithms and different kinds of data.

Recommender systems need to optimize the probability a member chooses an item and enjoys it enough to come back to the service. In order to do so, we should employ all the data that is available: from user ratings and interactions, to content metadata. More data availability enables better results. But in order to get those results, we need to have a framework that allows to experiment and measure the right metrics. This will allow us to take data-driven decisions that improve our customer experience, and grow our business value.

6. REFERENCES

[1] X. Amatriain. Mining large streams of user data for personalized recommendations. *ACM SIGKDD Explorations Newsletter*, 14(2):37–48, 2013.

[2] X. Amatriain, J. M. Pujol, and N. Oliver. I Like It... I Like It Not: Evaluating User Ratings Noise in Recommender Systems. In *User Modeling, Adaptation, and Personalization*, volume 5535, chapter 24, pages 247–258. Springer Berlin, 2009.

[3] R. M. Bell and Y. Koren. Lessons from the Netflix Prize Challenge. *SIGKDD Explor. Newsl.*, 9(2):75–79, December 2007.

[4] S. Funk. Netflix update: Try this at home. http://sifter.org/ simon/journal/20061211.html, 2006.

[5] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[6] R. Kohavi, A. Deng, B. Frasca, R. Longbotham, T. Walker, and Y. Xu. Trustworthy online controlled experiments: five puzzling outcomes explained. In *Proceedings of KDD '12*, pages 786–794, New York, NY, USA, 2012. ACM.

[7] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD*, 2008.

[8] Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD*, 2009.

[9] R. Salakhutdinov, A. Mnih, and G. E. Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proc of ICML '07*, 2007.