Performance of Recommender Algorithms on Top-N Recommendation Tasks

Gabriel Vargas Carmona

E-Business

Prof. Dr. Eduard Heindl

22 June 2012

Declaration.

I hereby certify that the research paper entitled "Performance of Recommender Algorithms on Top-N Recommendation Tasks" is my original work. I do realize that many references I quoted to support my paper.

____

Gabriel Vargas Carmona

Furtwangen, Germany 2012

Index

Increase of information

If we speak of the product marketing some years ago we will be submerged in a whole different world of information. Before the internet arrived there was a limited access to the information anybody can look when searching for a product. Publicity was the only way to make the product known in media such as music, radio or specialized magazines. And information that could be trusted was hard to get.

With the development of the Internet the situation is completely changed, from having limited access to the information to a world with virtually unlimited access to any type of marketing of a product. With the rapidly growing amount of information available on the web, tools are necessary to help users to select the relevant information. To satisfy this need, recommender systems have emerged. An example of these type of systems are: Netflix (movies), Amazon (books) and Last (music).

Recommender Systems

A recommender system is typically based in a set of users and a set of items. It works when each user "A" rates a subset of all the items with some numeric value (most common 1 to 5). The recommender system has to predict the unknown rating for source user "A" on a non-rated target item "Y" based on the known ratings. Collaborative filtering methods make recommendations based on the ratings of item "Y" by a set of users whose rating profiles are most similar to that of user "A" This is a effective method when talking about some ratings that have been analyzed to be common with ratings from other users, but it performs poorly for so-called cold start users, the ones that have expressed only very few ratings. This is due to the fact that cold start users, having only a few ratings, are unlikely to have users with similar rating profiles.

With the advent of online social networks, the trust-based approach to recommendation has emerged. This approach assumes a trust network among users and makes recommendations based on the ratings of the users that are directly or indirectly trusted by "A".

Trust-based recommenders can make recommendations as long as a new user is connected to a large enough component of the trust network. Therefore, trust-based methods tend to outperform collaborative filtering methods for cold start users.

Evaluation

A common practice with recommender systems is to evaluate their performance though error metrics such as RMSE between the actual ratings and the ratings predicted by the system.

Sometimes the system presents the "best bet", without taking into consideration the predicted rating values. That is, some items that are likely to be appealing for the user are suggested by the system.

RMSE and alternative methodologies.

Most of the methods used to analyze errors in the metrics are based on RMSE and MAE but they do not really measure top-N performance, they do approximations. Some alternative methodologies are presented based on accuracy metrics (recall and precision).

The paper is about the performance of collaborative filtering algorithms through the use of accuracy metrics for its evaluation finding the top-N recommendation task. It is important to say that the methods are contrasted with the RMSE metric and have been performed on the Netflix and Movielens datasets.

Testing Methodology (recall and precision)

We can understand that precision is the fraction of retrieved instances that are relevant for the user while recall is the fraction of relevant instances that are retrieved. This means then that both precision and recall are based on an understanding and measure of relevance. In other words, precision is a measure of exactness or quality, whereas recall is a measure of completeness or quantity.

For example, suppose a search engine returns 60 pages only 30 of which were relevant while failing to return 40 additional relevant pages. We can say then that the precision is 30/60 and the recall is 30/70..

Personalized vs non-personalized algorithms.

The first analysis revels that the non-personalized algorithms are comparable to the personalized algorithms. The first ones always show the same items (fixed list) but serve as baselines for more complex personalized algorithms. One of the aspects non-personalized algorithms is obsolete for is because users can get bored and disappointed of it.

There are two estimations of non-personalized algorithms:

- Movie Average, recommends top-N items with the highest average rating. The rating of user “A” in “Y” is predicted as the mean rating.
- Top Popular, recommends top-N items with the highest popularity. The rating cannot be inferred by the user.

When pursuing a top-N recommendation task exact rating prediction is not required. Some collaborative filtering algorithms that are no designed for minimizing RMSE, but consistently outperform other recommender algorithms in top-N recommendations are presented.

Long-Tail

When we are talking about the long-tail distribution, we understand that the majority of ratings are focused in a small fraction of the most popular items. Considering for the Netflix and Movielens datasets, figure 1 represents the distribution according to its long-tail behavior and popularity (having in the x axis the most popular at the bottom).

In the figure we can observe that about the 1.7% of the items (most popular) involve the 33% of ratings collected by Netflix (short head) while the 98% belong to the long-tail. We also note that Movielens’ rating distribution is slightly less long-tail, about the 5.5 % of the items represent the 33% of ratings.

[1]The really important issue is to recommend less known items more than known ones because the first ones can add serendipity to the users. To make a proper analysis of this part the graph each part, short head and long tail, should be studied separately.

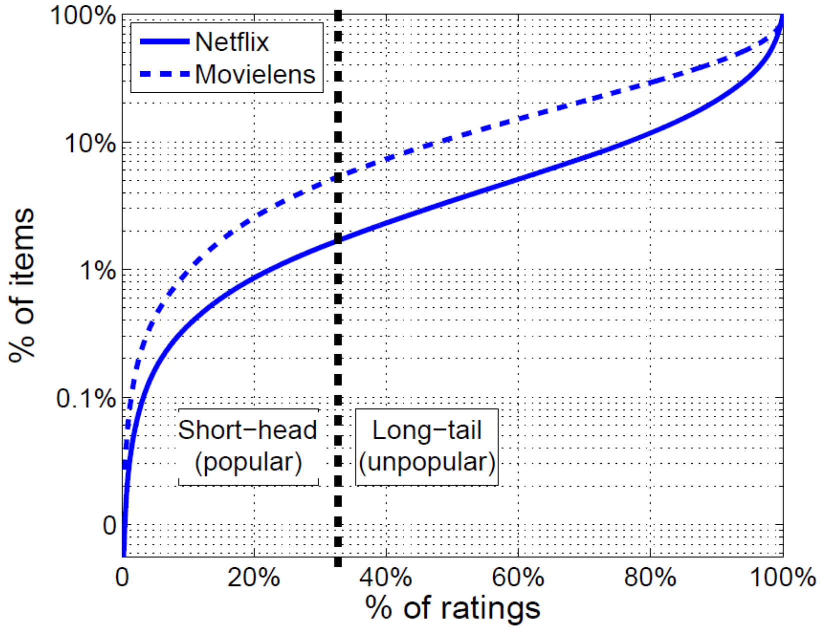


Figure 1: Rating distribution for Netflix (solid line) and Movilens (dashed line) datasets according to popularity.

Collaborative Algorithms

Most of the recommender systems are based on collaborative filtering (CF), where recommendations are based only on past user behavior (activities on items such as purchase, rental and clicks) using the pattern found to find a relation between users to users, items to items and finally users to items.

When analyzing these models there two type of approaches: neighborhood models and Latent factor. Each of them is explained next.

Neighborhood Models

This models base their prediction on the similarity of relationships among either users or items. They represent the most common approach to the CF. We can distinguish two types of algorithms: centered on user-user and centered on item-item similarity. The first ones predict the rating by user based on the ratings expressed by users similar to him about such item. The second ones have the user preference for an item based on his own ratings on similar items.

The neighborhood model is studied from the algorithm that is preferred and has better results, the one based on item-item similarity, thus, it has a better approach according to the RMSE and the relation can be explained in terms of other items previously rated by the user. Furthermore, this analysis brings perfect relation between the handling of users and ratings to the system.

The similarity between item *i* and item *j* is measured as the tendency of users to rate items *i* and *j* similarly. There are some cases when a sparse dataset is encountered; it is likely that some pairs of

[1] Graph taken from [1]

items have a poor support ending in a non-reliable measure. In that case a coefficient for shrinkage is defined. We can define the shrunk similarity as the following:

$$d_{ij} = \frac{n_{ij}}{n_{ij} + \lambda_1} s_{ij}$$

Where:

n_{ij} is the number of common raters
s_{ij} is the similarity between item I and j
d_{ij} is the shrunk similarity coefficient
λ_1 has a typical value of 100

Neighborhood models are further improved by means of KNN (k-nearest-neighborhood) approach. It decreases noise by the elimination of some items and improves the quality recommendations. Here are only considered the k items rated by “A” that are most similar to “Y”. This method also considers the biases that are more likely to mask the fundamental relation between items. Biases include item effects representing the fact that certain item tend to receive higher ratings and that some users tend to rate higher than others. Therefore, an item-item kNN method predicts the residual rating $(r_{ui} - b_{ui})$ as the residual ratings of similar items:

$$\hat{}r_{ui} = b_{ui} + \frac{\sum_{j \in D^k(u;i)} d_{ij}(r_{uj} - b_{uj})}{\sum_{j \in D^k(u;i)} d_{ij}}$$

NNCosNgbr (Non-normalized Cosine Neighborhood)

Considering that for top-N recommendation task an exact rating is not needed, items are rank simply by their appeal to the user. Therefore, the formula is simplified:

$$\hat{}r_{ui} = b_{ui} + \sum_{j \in D^k(u;i)} d_{ij}(r_{uj} - b_{uj})$$

It is important to mention that $\hat{}r_{ui}$ does not represent a proper rating, but is rather a metric for the association between user “A” and it “Y”

Latent Factor Models

They are formally known as the SVD models standing for Singular Value Descomposition. This type of models approaches model users and items as vectors. They have the use of matrix, and in the same space users and items are comparable; the rating of user “A” on item “Y” is predicted by the proximity between the related latent factor vectors.

The idea of the SVD models is to factorize the user-item rating matrix to a product of two lower rank matrices, user factor and item factor. Moreover, each user “A” is represented with a user actor vector $p_u \in R^f$. Similarly, each item “Y” is represented with an item factor vector $q_i \in R^f$. Prediction of a rating given by user “A” for item “Y” is computed as the product adjusted for biases

$$\hat{}r_{ui} = b_{ui} + p_u q_i^T$$

It is important to mention that SVD is undefined in the presence of unknown values such as missing ratings, therefore, several solutions as the estimation of a baseline are proposed. However for the standard method factorization becomes infeablesible.

These methods represent users as a combination of item features, without requiring any user-specific parameterization. In other words, they can create recommendations for users new to the system without re-evaluation of the parameters.

Pure SVD

While pursuing a top-N recommendation task, a correct item ranking is the objective without taking care about the exact rating prediction. For this type of method all missing values in the matrix are filled with zeros, a value with low importance. Then factorization can be done. The user rating matrix R is estimated by the next factorization:

$$R = U.\sum Q^T$$

Case of study

According to the case of study explained in [1], the quality of the datasets for MovieLens and Netflix are presented. The analysis is made by the use of the different methods explained here organizing the information into graphs.

It is important to mention that not every single method can be analyzed by the RMSE technique according to their characteristics (NNCosNgbr and PureSVD) however, those that present the necessary characteristics were improved by the use of this technique.

The analysis made according to the datasets is presented in some graphs shown below according to the company, the characteristics and explanation is presented further.

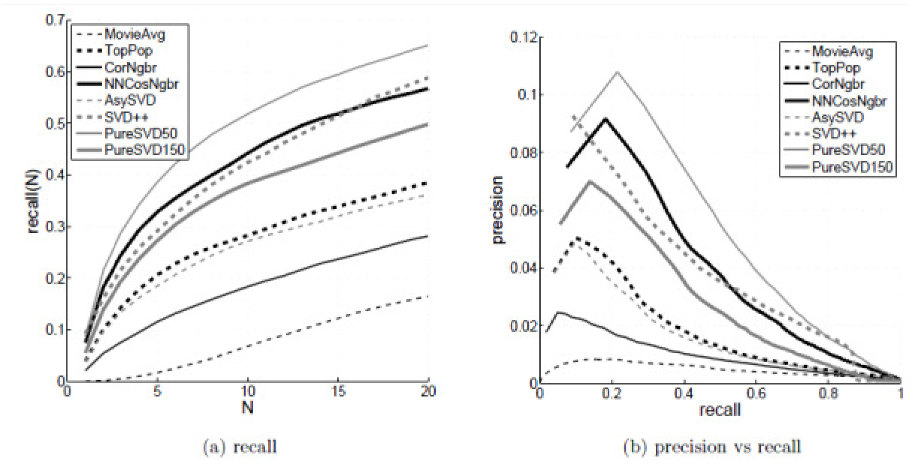
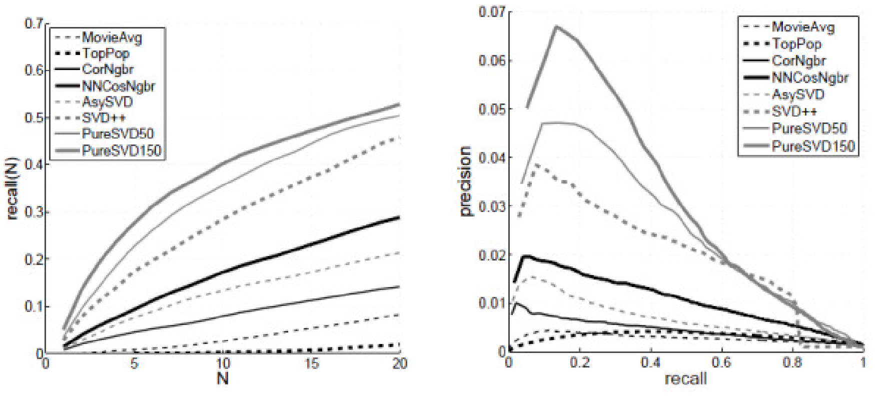
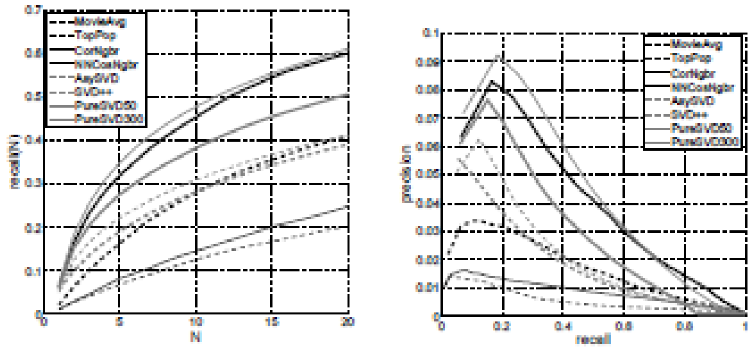


Figure 2: Movielens: (a) recall-at-N and (b) precision-versus-recall on all items. [2]

[2] Graph taken from [1]

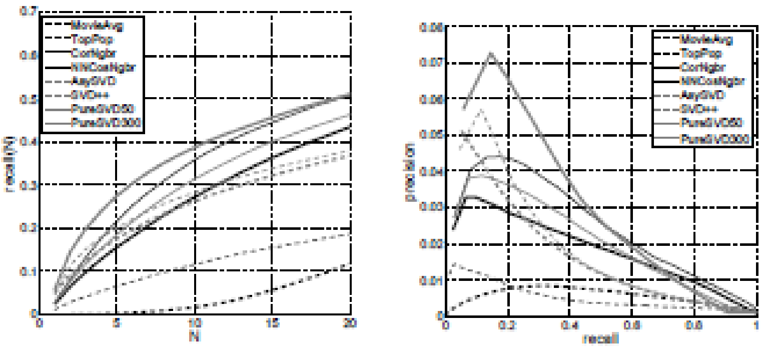


(a) recall (b) precision vs recall [3]



(a) recall (b) precision vs recall

Figure 4: Netflix: (a) recall-at-N and (b) precision-versus-recall on all items. [4][5]



(a) recall (b) precision vs recall

Figure 5: Netflix: (a) recall-at-N and (b) precision-versus-recall on long-tail (98% of items).

We can understand from all the graphics that the best result is when the pure SVD is applied it has the higher precision when the recall is the lowest. We have to consider some important factor here which is the bias created by the highest rated items, the comparison between the figures explain that growth in the graph.

[3] Graph taken from [1]
[4] Graph taken from [1]
[5] Graph taken from [1]

Analyzing each company separately we can see that MovieLens has greater precision than Netflix, in every case.

Conclusions

The way the recommender can be analyzed is based into accuracy metrics and error metrics were the last one is more popular. However to understand completely the behavior we need the accuracy metrics such as precision and recall.

There are several methods to get the best accuracy depending on the characteristics evaluated. It is important also to mention that top-n recommendations are really useful when managing the marketing of products that are not known by the people. These methods are focused on showing new products to users by relating the items to those they usually have in their environment.

The collaborative algorithm is the best way to understand the relation between items, users and both together.

To make an evaluation with higher accuracy we need to consider the top rated items and the bias they represent.

We have to consider also that the results given are only analyzed for this article, in order to have a more objective data more measurements should be made.

References

[1] P. Cremonesi, Y. Koren and R. Turrin. *Performance of Recommender Algorithms on Top-N Recommendation Tasks.* Page consulted on 15 June 2012. Available at: http://www.google.de/url?sa=t&rct=j&q=performance%20of%20recommender%20algorithms%20on%20top-n%20recommendation%20tasks&source=web&cd=1&ved=0CE4QFjAA&url=http%3A%2F%2Fwww.research.yahoo.net%2Ffiles%2Frecsys2010_submission_150.pdf&ei=tqnjT5nbIYjUsga6h-DFCQ&usg=AFQjCNFiOt8A6RYLMPYJ_02k2oWeYHhBwA

[2] S. M. Galán. *Filtrado Colaborativo y Sistemas de Recomendación*. Page consulted on 15 June 2012. Available at: http://www.it.uc3m.es/jvillena/irc/practicas/06-07/31.pdf

[3] M. Jamalí and M. Ester. *Using a Trust Network to Improve Top-N Recommendation*. Page consulted on 17 June 2012. Available at: http://www.cs.sfu.ca/~ester/papers/RecSys-2009-TopNRecommendation.final.pdf

[4] E notes. *Precision and recall.* Page consulted on 17 June 2012. Available at: http://www.enotes.com/topic/Precision_and_recall

Statsoft. *K-Nearest neighbors.* Page consulted on 17 June 2012. Available at: http://www.statsoft.com/textbook/k-nearest-neighbors/