Achieving Better Predictions with Collaborative Neighborhood

Edison Alejandro Garcia garcial@stanford.edu

Introduction

Collaborative Filtering (CF) is a popular method that uses machine learning to predict interest overlap between users based on their behavior such as common ratings of items, or common purchasing and browsing patterns.
Having an accurate algorithm can mean the difference between better automated predictions or hours of manual post-processing data. We propose a method that improves the quality of collaborative filtering approaches up to a factor of four percent from our baselines. Because of its modular design our method can be tweaked for higher quality or faster prediction. This flexibility makes it a good choice in a wide variety of practical applications.

Dataset

Our dataset of choice was Movielens which contains several dataset sources. From the latter we've chosen one dataset with 9000 movies and 700 users. We ran each algorithm on a fixed training and validation sets. We set out 30% of the total samples for the validation set while training on the remaining 70% of the original data.

Models

We defined two distinct approaches as our baselines:

- Average user-item:

$$\forall u,i\ X'(u,i) = \frac{\left(mean(X(u,:)) + mean(X(:,i))\right)}{2}, where\ X' \in \Re^{uxi}$$

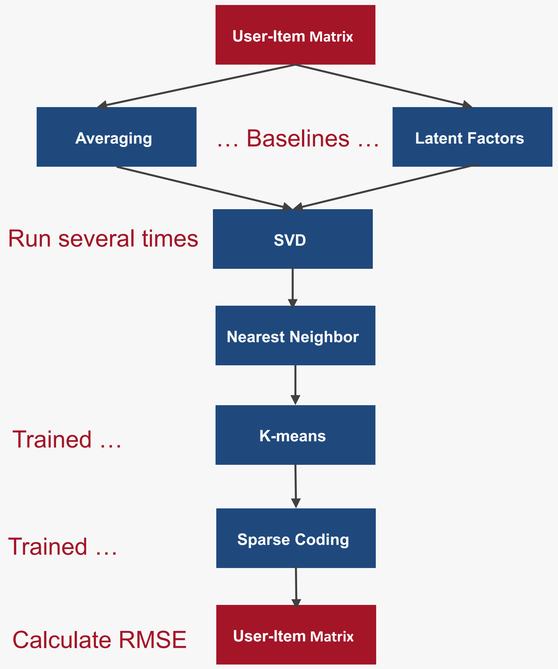
- Latent Factors with Biases:

$$U(u,i) = \mu + b_u + b_i, \forall u,i \in X\ and\ \{u,i \mid X(u,i) = nil\}$$

On top of the baselines we run a set of models based on collaborative neighborhood:

- Singular Value Decomposition (SVD)
- K-means (KM)
- Nearest-Neighbor (NN)
- Sparse Coding (SC)

Experiments



Results

Pipeline	Results (RMSE)
Averaging (A)	0.945
Latent Factors (LF)	1.014
A+(8x)SVD+NN+KM+SC	0.905
LF+(8x)SVD+NN+KM+SC	0.979
A+(8x)SVD+KM	0.914
LF+(8x)SVD+KM	0.989

The table present the results for both baselines and the combinations that presented the lowest errors.

Discussion

The results present a substantial improvement over the baseline, mainly due to SVD. We consider that having the best possible outcome is absolutely essential even at a higher time cost[2]. We actually disagree in some context because best possible outcomes do not translate into better predictions. Therefore we've choose speed over lower RMSE.
For these highly sparse data matrices we need to have a more advanced cold start. It is not enough to just calculate averages because they do not account for differences across data points.
A further benefit of our approach is it simplicity and highly configurable depending on applications requirements.

Challenges

- Achieving high coverage with a sparsity in the dataset at 98.5%.
- We found some differences between datasets results due to overfitting. With the large amount of parameters, performing cross validation proved to be time consuming.
- How to produced high quality recommendations? Regardless of having a low RMSE, we can still have low prediction quality. For the latter more data or less sparsity would be a good start.
- Low computational power over large datasets.

Future

- In the short term we need to fix the cold start by providing a solution to better initialize missing values with our two baselines.
- Add a new set of models proven to be computationally efficient and with low error of margins.

Main References

[1] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context", 2015
[2] B. S. Francesco Ricci, Lior Rokach and P. B. Kantor, *Recommender Systems Handbook*, 2011
[3] M. H. Pryor, "The Effects of Singular Value Decomposition on Collaborative Filtering,"