Eigenvalue analogy for confidence estimation in item-based recommender systems

2 authors:

Maurizio Ferrari Dacrema
Politecnico di Milano
12 PUBLICATIONS   **19** CITATIONS

SEE PROFILE

Paolo Cremonesi
Politecnico di Milano
160 PUBLICATIONS   **2,333** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Video Recommendation by Leveraging Multimedia Content View project

Confidence estimation View project

Eigenvalue analogy for confidence estimation in item-based recommender systems

Maurizio Ferrari Dacrema
Politecnico di Milano, Italy
maurizio.ferrari@polimi.it

Paolo Cremonesi
Politecnico di Milano, Italy
paolo.cremonesi@polimi.it

ABSTRACT

Item-item collaborative filtering (CF) models are a well known and studied family of recommender systems, however current literature does not provide any theoretical explanation of the conditions under which item-based recommendations will succeed or fail.

We investigate the existence of an ideal item-based CF method able to make perfect recommendations. This CF model is formalized as an eigenvalue problem, where estimated ratings are equivalent to the true (unknown) ratings multiplied by a user-specific eigenvalue of the similarity matrix. Preliminary experiments show that the magnitude of the eigenvalue is proportional to the accuracy of recommendations for that user and therefore it can provide reliable measure of confidence.

KEYWORDS

Recommender confidence, Inverse eigenvalue, Item-based CF

1 RELIABILITY ESTIMATION

Item-item Collaborative Filtering (CF) is commonly regarded as one of the most effective approaches to build Recommender Systems. Given a set of users, a set of items and a set of ratings, item-based CF techniques use available ratings to build relationship between items in the form of an item *similarity matrix*.Although research on item-based algorithms is not new, the evaluation of their predictive capabilities is purely empirical and still an open issue. In most works, predictions are expected to be more accurate for items and users associated with many ratings and low variance of ratings. Adomavicius et al. [1] observe that recommendations tend to be more accurate for users and items exhibiting lower rating variance. Cremonesi et al. [2] empirically show that there is some correlation between recall of recommendations and number of ratings in the user profile. A disadvantage of these approaches is that confidence estimation is based only on user and item ratings, and does not take into account the properties of the prediction model, which could overlook some valuable information.

We show that, under the hypothesis of a *perfect model* (i.e., an oracle able to make perfect recommendations), item-based methods are analogous to an *eigenvalue problem*, where each user profile (i.e., the vector of ratings from a user) is a left eigenvector of the similarity matrix. Moreover, each user is associated with an eigenvalue, and the magnitude of the eigenvalue is linearly correlated with the accuracy of recommendations for that user. We call this analogy the *eigenvector analogy*.

2 EIGENVECTOR ANALOGY

A generic item-based model (either CF, CBF or hybrid) predicts the ratings $\tilde{\mathbf{r}}_u$ of a user u for all items as:

$$\mathbf{r}_u \mathbf{S} = \tilde{\mathbf{r}}_u \tag{1}$$

where $\mathbf{r}_u$ is the profile of the user and $\mathbf{S}$ is a similarity matrix. We now assume to have an ideal item-based recommender, able to predict all the user ratings (both known and unknown). More formally, this translates into two assumptions:

Assumption 1: Perfect Predictions. Estimated ratings $\tilde{\mathbf{r}}_u$ are identical to the ratings in the user profile $\mathbf{r}_u$. Because of this assumption, we can write $\tilde{\mathbf{r}}_u = \mathbf{r}_u$.

Assumption 2: Perfect Knowledge. We have the complete knowledge of all the ratings in the user-rating matrix.

Under these assumptions, the item-based model described by (1) is analogous to a left eigenvector problem

$$\mathbf{r}_u \mathbf{S} = \lambda_u \mathbf{r}_u \tag{2}$$

where $\mathbf{r}_u$ is a **left** eigenvector of matrix $\mathbf{S}$ and λ_u is the corresponding eigenvalue. In the eigenvector formulation (2), predicted ratings are equivalent to the ratings in the user profile *multiplied by the eigenvalue associated with the user*. For each user, the corresponding eigenvalue can either flatten out or amplify the differences between predicted ratings. The closer is an eigenvalue to zero, the more difficult will be for the item-based method to correctly rank items and to distinguish between relevant and non-relevant items.

2.1 Learning the eigenvalues

In order to provide an estimation of the user's eigenvalue, we rewrite (2) in matrix format

$$\mathbf{R}\mathbf{S} = \mathbf{\Lambda}\mathbf{R} \tag{3}$$

where $\mathbf{R}$ is the user-rating matrix, $\mathbf{S}$ is the similarity matrix and $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues on its diagonal. In order to satisfy the model described by (3), we need to find a similarity matrix $\mathbf{S}$, other than the identity matrix, such that all the user profiles $\mathbf{r}_u$ in $\mathbf{R}$ are left eigenvectors of $\mathbf{S}$. This problem is called the *inverse eigenvalue problem* which, if we also know all the eigenvalues, has
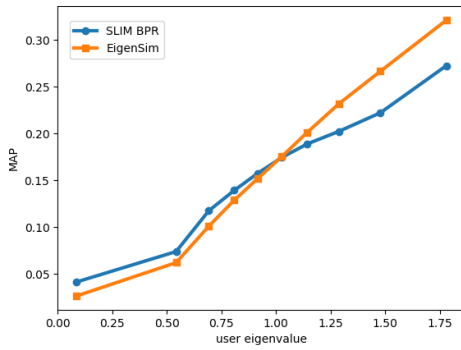
Figure 1: EigenSim and Slim MAP@5 for increasing values of the threshold on λ_u, on the Movielens 10M dataset.

Figure 2: Eigenvalue λ_u vs. number of ratings in the user profile for the Movielens 10M dataset.

the following exact solution: $\mathbf{S} = \mathbf{R}^+ \mathbf{\Lambda} \mathbf{R}$. Where $\mathbf{R}^+$ is the Moore-Penrose pseudoinverse, which uses singular-value decomposition. We call this model EigenSim. The eigenvalues are then estimated via SGD, optimizing BPR.
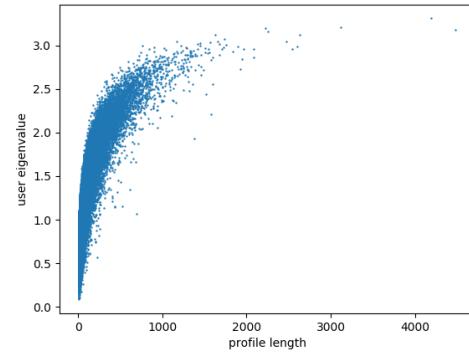
3 DATASETS

We evaluated the performance of EigenSim on two different datasets, namely Movielens 10M (70K users, 10M interactions) and subsample of Netflix (40K users, 1.25M interactions), the dataset used for the Netflix Prize. The datasets are split by selecting randomly ratings into training (60%), validation (20%) and test (20%) set.

4 RESULTS DISCUSSION

The main focus of our experiment is to study if there is a correlation between the eigenvalue associated with a user and the quality of recommendations it receives, measured in terms of MAP. We also investigate if this correlation depends on the specific algorithm used for recommendations. Moreover, as previous work suggested, we study if there is correlation between the user's number of ratings and his/her eigenvalue. Figure 1 presents the performance of EigenSim and Slim on the Movielens 10M dataset. The 70 thousands users of the dataset have been ranked based on their eigenvalue λ_u and partitioned into 10 groups of 7K users each. Each point in the figure represents the MAP for each group. The corresponding eigenvalue is the median of the eigenvalues of the users in the group. The figure clearly shows a linear correlation between the quality of recommendations and the eigenvalue of the users. When providing recommendations to users with very low eigenvalues the quality of recommendations drops to zero, regardless of the algorithm, while for users with large eigenvalues the quality of recommendations increases for both algorithms. In terms of MAP, the quality of ItemKNN recommendations for users with large eigenvalues ($\lambda_u = 1.75$) is almost ten times the quality for users with low eigenvalues ($\lambda_u = 0.1$) and twice the average quality across all the users. It is interesting to observe that eigenvalues affect the quality of recommendations also for an item-based algorithm (Slim) not based on the eigenvalue assumption.

Pearson correlation coefficient between MAP and λ_u is 0.99 for both Slim and EigenSim. This confirms that eigenvalues are

strong predictors of the quality of recommendations for any item-based algorithms. For comparison, on the same dataset, correlation between MAP and profile length is only 0.78 and 0.80 for Slim and EigenSim, respectively. A similar behaviour holds for Netflix.

Figure 2 plots, for each user u in the Movielens 10M dataset, the eigenvalue associated with the user, as computed with EigenSim, and the number of ratings in his/her profile. The figures show that there is not a strong correlation between λ_u and the number of ratings in the user profile. For instance, users with a profile length of 50 ratings have eigenvalues ranging from 0.3 (very low quality of recommendations) up to 1.0 (average quality). The Pearson correlation coefficient between λ_u and profile length is 0.76. The correlation between quality of recommendations and eigenvalues is much stronger than the correlation between quality of recommendations and number of ratings in the user's profile.

These results are not limited to item-based methods but apply also to some model-based matrix factorization methods, such as AsySVD and PureSVDas, thanks to folding-in, matrix factorization models are equivalent to the FISM item-based method [3].

5 CONCLUSION

We have shown that an ideal item-based method can be formulated as an eigenvalue problem. We show that the magnitude of the eigenvalue is strongly correlated to the accuracy of recommendations for that user and therefore it can provide a reliable measure of confidence. Ongoing work is focused on providing a more in-depth theoretical analysis of the eigenvalue analogy and its extension to user-based methods, as well as to validate these results with more datasets and compare against other proposed confidence measures.

REFERENCES

[1] Gediminas Adomavicius, Sreeharsha Kamireddy, and YoungOk Kwon. 2007. Towards more confident recommendations: Improving recommender systems using filtering approach based on rating variance. In *17th Workshop on Information Technologies and Systems, WITS 2007*.

[2] Paolo Cremonesi, Franca Garzottto, and Roberto Turrin. 2012. User Effort vs. Accuracy in Rating-based Elicitation. In *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12)*. 27–34.

[3] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 39–46.