The Effect of Sparsity on Collaborative Filtering Metrics

Jesus Bobadilla & Francisco Serradilla
Universidad Politecnica de Madrid
Crta. De Valencia, Km 7, 28031 Madrid, Spain

jbobi@eui.upm.es, fserra@eui.upm.es

Abstract

This paper presents a detailed study of the behavior of three different content-based collaborative filtering metrics (correlation, cosine and mean squared difference) when they are processed on several ratio matrices with different levels of sparsity. The total number of experiments carried out is 648, in which the following parameters are varied: metric used, number of k-neighborhoods, sparsity level and type of result (mean absolute error, percentage of incorrect predictions, percentage of correct predictions and capacity to generate predictions). The results are illustrated in two and three-dimensional representative graphs. The conclusions of the paper emphasize the superiority of the correlation metric over the cosine metric, and the unusually good results of the mean squared difference metric when used on matrices with high sparsity levels, leading us to interesting future studies.

Keywords: recommender systems, sparsity, collaborative filtering, metric

1 Introduction

At present, Recommender Systems (RS), are broadly used to implement Web 2.0 services (Janner 2007) as mentioned by Knights and Lin (2007), based on Collaborative Filtering (CF). RS make predictions about the preferences of each user based on the preferences of a set of “similar” users.

This way, a trip to Canary Islands could be recommended to an individual who has rated different destinations in the Caribbean very highly, based on the positive ratings about the holiday destination of “Canary Islands” of an important number of individuals who also rated destinations in the Caribbean very highly.

There are a large number of applications based on RS (Jinghua 2007, Baraglia 2004, and Fesenmaier 2002), some of which are centered on the movie recommendation area (Konstan 2004, Antonopoulus 2006, Li 2005).

The quality of the results offered by a RS greatly depends on the quality of the results provided by its CF (Adomavicius 2005, Herlocker 2004) phase; i.e. it is essential to be capable of adequately selecting the group of users most similar to a given individual.

The similarity among users can be computed in three different ways: content-based methods, model-based methods and hybrid approaches. Content-based methods (Breese 1998, Kong 2005) use similarity metrics (Herlocker 2004) which operate directly on the individual user’s ratios (in the trip recommender example, that is each value voted for each travel destination). Model-based methods (Breese 1998) use user ratios to create a computable model (Bayesian classifier (Cho 2007), neural network (Ingoo 2003), fuzzy system [16], etc.) and from this model they predict the clusters of similar users.

At present, for reasons of predictability and efficiency, commercial RS (Linden 2003) are implemented using content-based CF metrics. Model-based CF can usually be found in non-commercial research phases.

The majority of CF research aims to increase the accuracy and coverage (Giaglis 2006, Li 2005, Fuyuki 2006, and Manolopoulus 2007); nevertheless, it is advisable to improve certain other factors: effectiveness of recommendations, searching for good items, credibility of recommendations, precision and recall measures, etc.).

Memory-based methods work on two-dimensional matrices of U users who have rated a number of items I. We can consider a RS running in an e-travel agency, where, over the years, thousands of travelers have rated hundreds of destinations, for example.

An important problem in obtaining effective predictions using RS is the fact that most of the users only rate a very small proportion of the items; this is known as the sparsity problem. When the matrix is very sparse, it means there are many users who have rated very few items and this leads to two main negative effects:

- The set of similar users (k-neighborhoods) (Herlocker 2002) does not suitably match the preferences of the recommended user (there are not enough common rated items to establish a reliable similarity result between two users).
- It is not easy to recommend items to the user, as you are not likely to find enough k-neighborhoods who had rated the same items positively.

Consequently, the accuracy and the majority of the main effectiveness measures of the CF predictions drop when they are applied to extremely sparse matrices, leading to the users losing confidence in the RS service as a whole.

The sparsity problem has traditionally been tackled using user profile information to reinforce the similarity measure. The CF techniques called demographic filtering (Pazzani 1999) use all the possible additional information to establish the similarity among users such as gender, age, education, area code, etc.

Another approach in order to reduce the sparsity problem is the use of a dimensionality reduction technique such as Singular Value Decomposition (SVD) (Sarwar 2000).

The demographic filtering approach has two important restrictions:

- More often than not there is no demographic information (or not enough demographic information) in the RS database.
- Establishing similarities based on demographic information is very risky and can easily lead to incorrect recommendations.

The dimensionality reduction approach removes unrepresentative users or items. At present some research works use statistical techniques such as Principle Component Analysis (PCA) (Goldbergh 2001) and information retrieval techniques such as Latent Semantic Indexing (LSI) (Deerwester 1990, and Hofmann 2003). The main problem with the reduction approach is the inherent loss of information in the reduction process.

Alternatively, other approaches exist with which to deal with the sparsity problem, such as the use of trust inferences (Papagelis 2005), attraction-weighted information filtering (Bruyn 2004) and topographic organization of user preferences patterns (Polcicova 2004).

2 Content-Based Metrics

Content-based methods work on a table of U users who have rated a number of items I. The prediction of a non-rated item i for a user u is computed as an aggregate of the ratings of the K most similar users (k-neighborhoods) for the same item i, where $\tilde{K}$ denotes the set of k-neighborhoods.

The most common aggregation approaches are the average (1) and the weighted sum (2).

$$r_{u,i} = \frac{1}{|\tilde{K}|} \sum_{k \in \tilde{K}} r_{k,i} \qquad (1)$$

$$r_{u,i} = \mu \sum_{k \in \tilde{K}} sim(u,k)\, r_{k,i} \qquad (2)$$

Where μ acts as a normalizing factor, usually computed as:

$$\mu = 1 \Big/ \sum_{k \in \tilde{K}} sim(u,k) \qquad (3)$$

The similarity approaches usually compute the similarity between two users *x* and *y: sim(x,y)* based on their ratings of items that both users have rated (4).

$$i \in I \mid r_{x,i} \neq \phi \ and \ r_{y,i} \neq \phi \qquad (4)$$

The most popular similarity metrics are Pearson correlation (5) and cosine (6), although we will complete the experiments in this paper by adding the least known Mean Squared Difference (MSD) metric (7).

$$sim(x,y) = \frac{\sum_i (r_{x,i} - \bar{r}_x)(r_{y,i} - \bar{r}_y)}{\sqrt{\sum_i (r_{x,i} - \bar{r}_x)^2 \sum_i (r_{y,i} - \bar{r}_y)^2}} \qquad (5)$$

$$sim(x,y) = \frac{\sum_i r_{x,i}\, r_{y,i}}{\sqrt{\sum_i r^2{}_{x,i}} \sqrt{\sum_i r^2{}_{y,i}}} \qquad (6)$$

$$sim(x,y) = \frac{1}{I} \sum_{i=1}^{I} (r_{x,y} - r_{y,i})^2 \qquad (7)$$

The research work shown in this paper is based on comparative experiments using Pearson (5), cosine (6) and MSD (7) metrics, the average aggregation approach (1), and the Mean Absolute Error (8).

MSD has been selected due to its unique behavior, which is very different to the correlation and cosine metrics, mainly when it is used in sparse ratio matrices.

3 Design of Experiments

In order to discover the behavior of each of the three metrics analyzed, we used the *MovieLens* database [1] as a base, which has been a reference for many years in research carried out in the area of CF.

The database contains 943 users, 1682 items and 100,000 ratings, with a minimum of 20 items rated per user. The items represent cinema films and the rating ranges vary from 1 to 5 stars.

In all the experiments carried out, for each film that each user has rated, the average value of the ratios given by their k-neighborhoods for that film has been calculated and the prediction has been compared with the value rated by the user, thus obtaining the calculation of the mean absolute error (MAE) [8].

[1] Our acknowledgements to the GroupLens Research Group

$$\left|\overline{E}_{user}\right| = \frac{\sum_{i=1}^{I}\left|\frac{1}{\tilde{K}}\sum_{k\in\tilde{K}} r_{k,i} - r_{user,i}\right|}{I} \qquad (8)$$

The previous process was carried out for each of the following k-neighborhoods values: 15, 30, 60, 90, 120, 150, 180, 210 and 240, covering from 1.6% to 25.4% of the total number of users.

In order to obtain comparable results based on different sparsity levels, we have made several reductions on the original database containing 100,000 ratings; each reduction process has removed a fixed number of database ratios using a random function. In this way, we have obtained five additional databases: 80,000 ratings, 60,000 ratings, 40,000 ratings, 20,000 ratings and 10,000 ratings.

The MovieLens original database presents a $100-10^7/(943*1682)$ percentage of sparsity, the 80,000 database presents a $100-8*10^6/(943*1682)$ percentage of sparsity, and so on. Therefore, the sparsity range covered in the experiments is: 93.7%, 94.96%, 96.22%, 97.48%, 98.74% and 99.37%.

In turn, all of these calculations have been carried out 3 times (one for each metric included in the study).

The total number of experiments carried out is 648 (9 k-neighborhoods * 6 sparsity levels * 3 metrics * 4 types of results).

The experiments have been grouped in such a way that the following can be determined:

- Accuracy.
- Number of recommendations made.
- Number of perfect predictions.
- Number of bad predictions.

We consider a perfect prediction to be each situation in which the prediction of the number of stars recommended for one user in one film matches the value rated by that user for that film.

We consider a bad prediction to be each situation in which the prediction of the number of stars recommended for one user in one film is different by more than 2 stars from the value rated by that user for that film.

We consider a recommendation made to be each situation in which a user has rated an item and at least one of the user's k-neighborhoods has also rated it, in such a way that a prediction could be made and an MAE error obtained.

4 Results

The results section has been divided into six subsections: the first one shows comparatives of the three metrics studied, processed using different levels of sparsity; in this case no detailed information on k-neighborhoods is included as each result (each graph) has been obtained by calculating the average of the individual results of all nine (15 to 240) k-neighborhoods.

The remaining three subsections refer to each one of the three metrics, respectively, and they contain all the detailed information obtained when processing all the possible variations: range of k-neighborhoods / range of sparsity levels.

The last subsections illustrate the details obtained by comparing the correlation metric with the cosine and the MSD metrics.

4. 1 Comparison of CF Metrics using different sparsity levels

The first results presented here refer to accuracy, processed using the Mean Absolute Error. The x-axis represents the different percentages of sparsity.

Figure 1 shows better results with the correlation metric than with the cosine metric. In fact, there is an improvement in the correlation results, in contrast to the cosine results, where the error increases as the sparsity percentage increases.

The Mean Squared Difference metric shows much better results than the cosine and correlation metrics; nevertheless it is necessary to adjust this good result with the very poor behavior obtained in Figure 2.

The MAE values indicate the mean absolute difference between predictions and real rated values, consequently, a value of 0.5 on the MAE axis means a half-star error in the values predicted from the 'MovieLens' database.

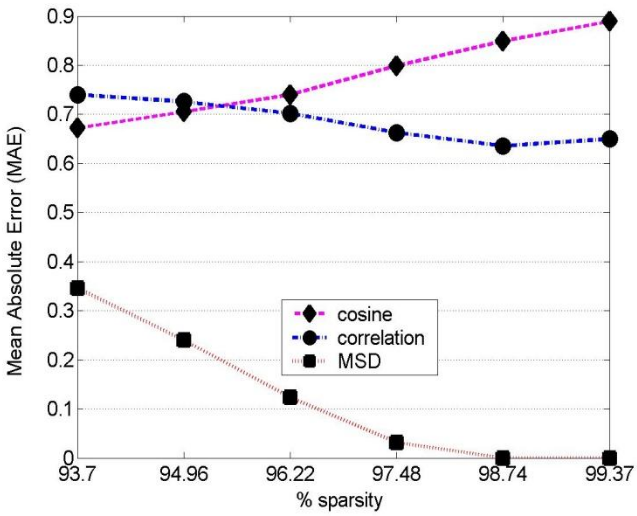

Figure 1. Mean Absolute Error

Figure 2 shows the percentage of recommendations that each metric is able to produce. These percentages are obtained by dividing each number of predictions obtained by computing the different levels of sparsity by the number of ratios of each database (10,000, 80,000, etc.), for example, using the 10,000-ratio database (99.37% of sparsity) the cosine metric was able to compute an average of 8688 predictions, thus the last diamond position in Figure 2 has the value 86.88%.

The correlation metric once again shows better behavior than the cosine metric as it is able to obtain a larger number of recommendations; nevertheless, as expected, the amount of predictions decreases as the sparsity level increases (it is more difficult to obtain recommendations when the quantity of information available to make predictions decreases).

The rising section of the cosine function in Figure 2 can be explained by the erroneous behavior of the cosine metric when the sparsity of the vectors is too high, which is the case when the sparsity of the database is high.

The MSD metric provides very poor results as it obtains a low quantity of predictions. This is this metric's Achilles' heel and is the aspect that should be improved in any metric derived from it.

Figure 3 shows that the correlation metric is able to achieve a greater number of prediction hits than the cosine metric. Whereas the cosine hits drop in line with the sparsity level, the correlation metric even manages to improve its results when the percentage of sparsity is high.

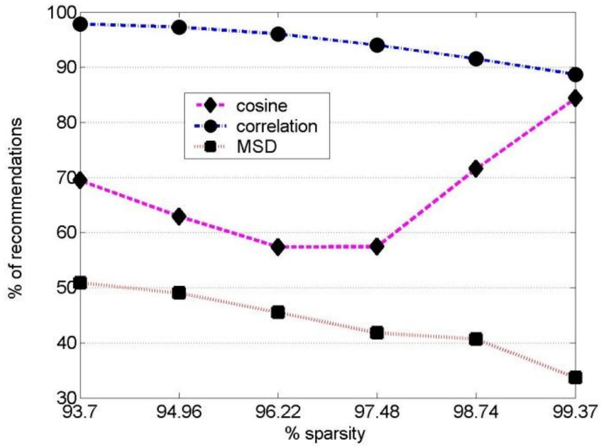

Figure 2. Percentage of recommendations made

It is important to realize that the cosine metric improves the MAE and the percentage of perfect prediction results compared to the correlation metric when the sparsity percentage is not very high (database of 100,000 ratings).

The MSD exhibits excellent behavior when the sparsity levels are high; nevertheless, it is important to realize that the overspecialization effect (recommending items that are too well-known) can be easily produced.

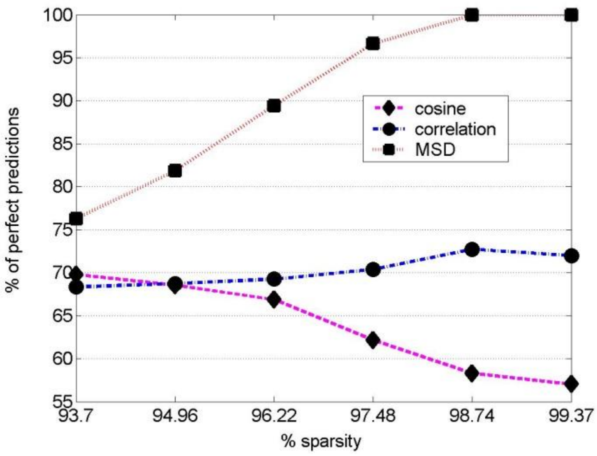

Figure 3. Percentage of perfect predictions

A very important objective of CF metrics is to avoid incorrect recommendations, to prevent users from losing confidence in the system.

Figure 4 presents the percentage of incorrect recommendations (more than two stars of difference between predictions and real ratios). As we can see, the cosine metric does not respond well to an increase in sparsity, whereas the correlation metric responds well. The MSD metric does not produce a high quantity of predictions (Figure 2), but it appears to achieve a good number of hits with its recommendations (Figures 3 and 4), particularly when the sparsity levels are high.

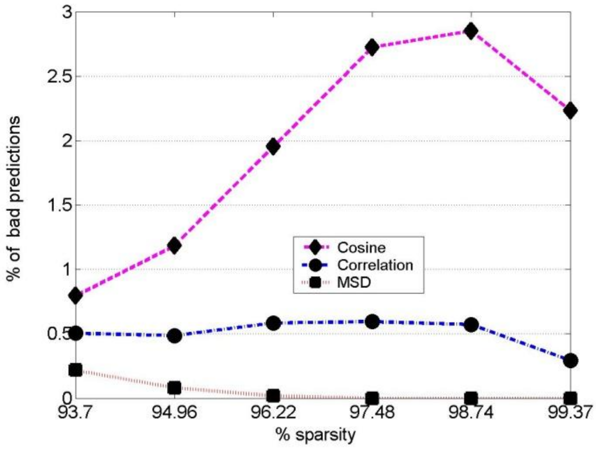

Figure 4. Percentage of bad predictions

4.2 Correlation Metric

This section shows the detailed results obtained from the correlation metric experiments. As in the previous section, the aspects of study are: MAE accuracy, percentage of recommendations made, percentage of perfect predictions and percentage of bad predictions.

Each result is presented as a three-dimensional graph where the *x-axis* represents the number of k-neighborhoods computed in each experiment and the *z-axis* represents the percentage of sparsity (i.e. the 100,000, 80,000, … databases used).

Figure 5a) shows an even decline of the MAE when the sparsity percentage increases (as we saw in Figure 1). In this case we can observe that correlation works better when the number of k-neighborhoods is not low.

Figure 5b) shows the poorest results when the number of k-neighborhoods is low (less than 60). The evolution presented in Figure 2 would improve by selecting more than 60 k-neighborhoods. The same is true when the correlation metric obtains perfect predictions (Figure 5c) and bad predictions (Figure 5d). As a result of this, we can highlight the good results obtained by this metric, especially when the number of neighborhoods is not low and the sparsity level is high.

Figure 5c) shows how the rising correlation slope presented in Figure 3 can be enhanced by selecting a number of k-neighborhoods higher than 120.

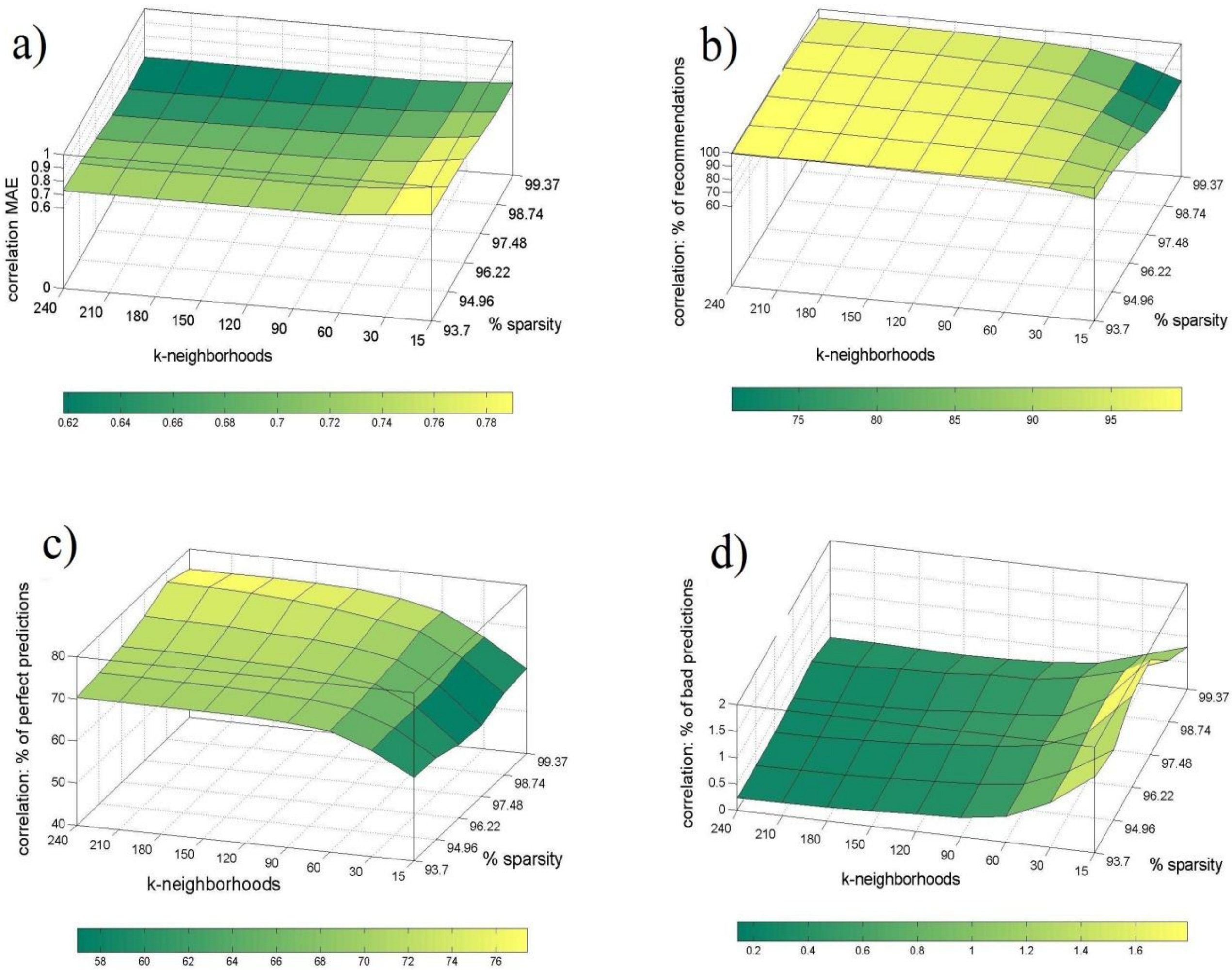

Figure 5. Results of the correlation metric: a) Mean Absolute Error, b) percentage of recommendations made, c) percentage of perfect predictions, d) percentage of bad predictions

4.3 Cosine Metric

Although it was previously established that the correlation metric presented better behavior than the cosine metric, it is relevant to point out the details of the cosine, which is much less regular than the Pearson metric.

In general, it can be said that the cosine metric works better when the sparsity level is low and the number of k-neighborhoods is high. This fact can be observed in Figure 6, where the best results are given in the quadrant: k-neighborhoods from 150 to 240 and sparsity from 93.7 to 96.22.

Figure 6a) shows the best results when the values are smaller (from 0.65 to 0.75); the same situation is presented in Figure 6d) (from 0.5 to 1.5). Figures 6b) and 6c) give the best results when the values are larger (from 80 to 100 and from 68 to 78, respectively).

By studying Figures 6a) to 6d) (cosine) we can observe that the slopes of the surfaces are higher than those corresponding to Figures 5a) to 5d) (correlation), both on the sparsity axis and the k-neighborhoods axis (when k>60); this means that the influence of both parameters is higher in the cosine metric.

4.4 Mean Squared Differences Metric

The results obtained by applying the MSD metric are significantly different to those obtained by the cosine and correlation metrics. The mean absolute error (Figure 7a) presents very low (good) values for all the k-neighborhoods and the percentage of sparsity ranges. We can observe that the best results (lowest errors) are obtained by selecting the lowest k-neighborhood values and, particularly, when the sparsity percentage is high.

There can be no doubt that the weak point of the MSD metric is its poor capacity to generate a large

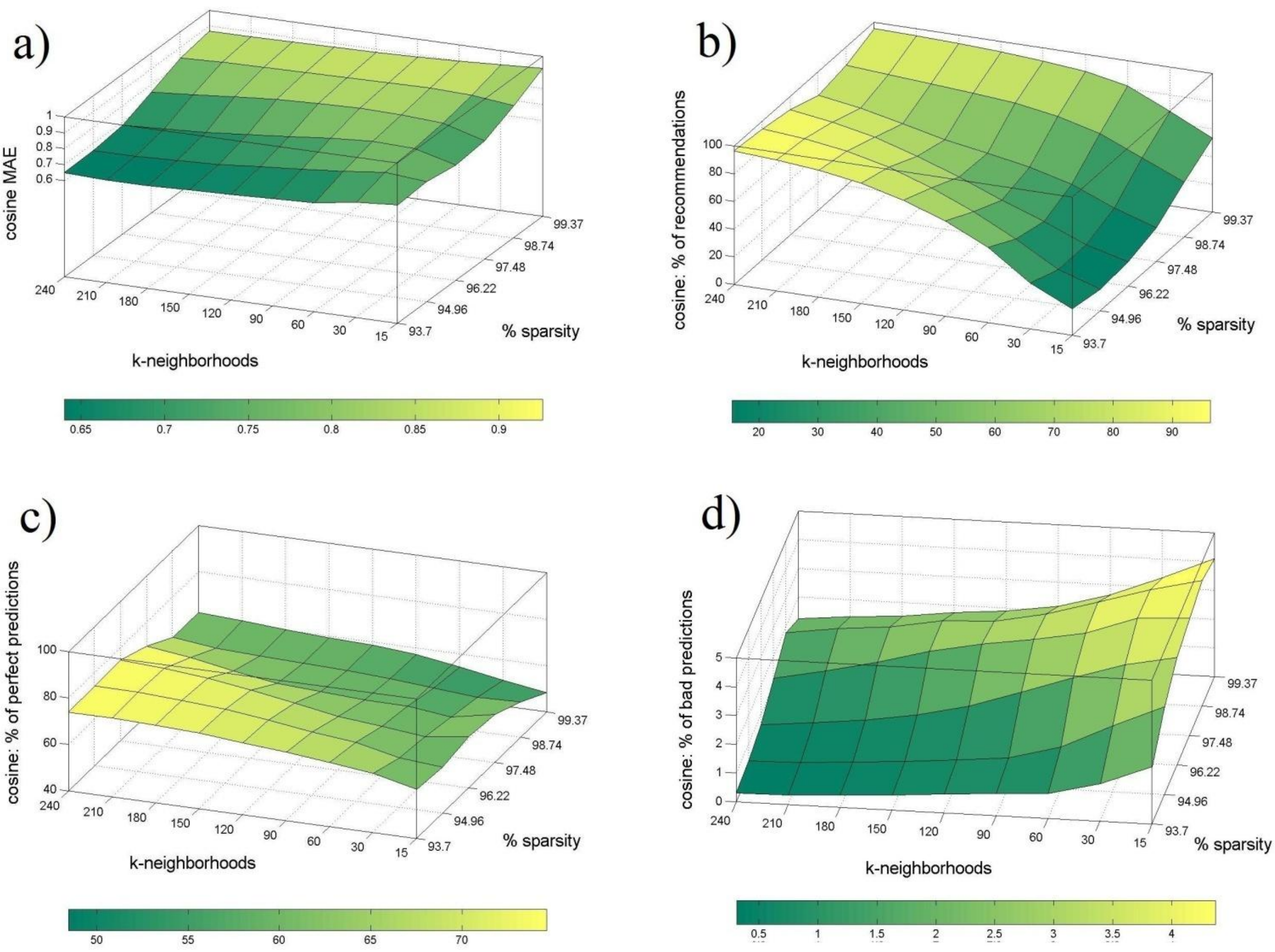

Figure 6. Results of the cosine metric: a) Mean Absolute Error, b) percentage of recommendations made, c) percentage of perfect predictions, d) percentage of bad predictions

number of predictions. As can be seen in Figure 7b), the percentage of recommendations obtained using the MSD metric is lower than that obtained using the cosine metric and even more so in the case of the correlation metrics. In this aspect, it can be observed that the function's slope is much more significant in the k-neighborhoods axis than in the percentage of sparsity axis; therefore, this aspect can be improved by choosing a high k-neighborhood value as opposed to working with high sparsity databases.

The quality of the recommendations (measured as high levels of perfect predictions combined with low levels of bad predictions) is very good when using the MSD metric, in comparison to the cosine and correlation metrics; this is mainly due to the low percentage of bad recommendations. By observing Figures 7c) and 7d) it can be determined that the best results (more perfect predictions and fewer bad predictions) are obtained at the highest values of sparsity. The poorest results occur when the highest k-neighborhood values are combined with the lowest percentages of sparsity levels.

In short, when using the MSD metric with low values of sparsity, it is necessary to choose a suitable k-neighborhood value to obtain a balance between quality (Figures 7a,c,d) and capacity to recommend (Figure 7b); the highest values of the k-neighborhood parameter offer us a better capacity for recommendation, while the lowest values of the k-neighborhood parameter lead to an improvement in the quality.

The most interesting observation in Figure 7 is that all the objectives (low error, high capacity to recommend, high percentage of perfect predictions and low percentage of bad predictions) are improved at the same time when the sparsity value increases. This characteristic confers a special importance to the MSD metric to be used in very sparse RS databases and it opens a way to creating new specialized MSD-based metrics.

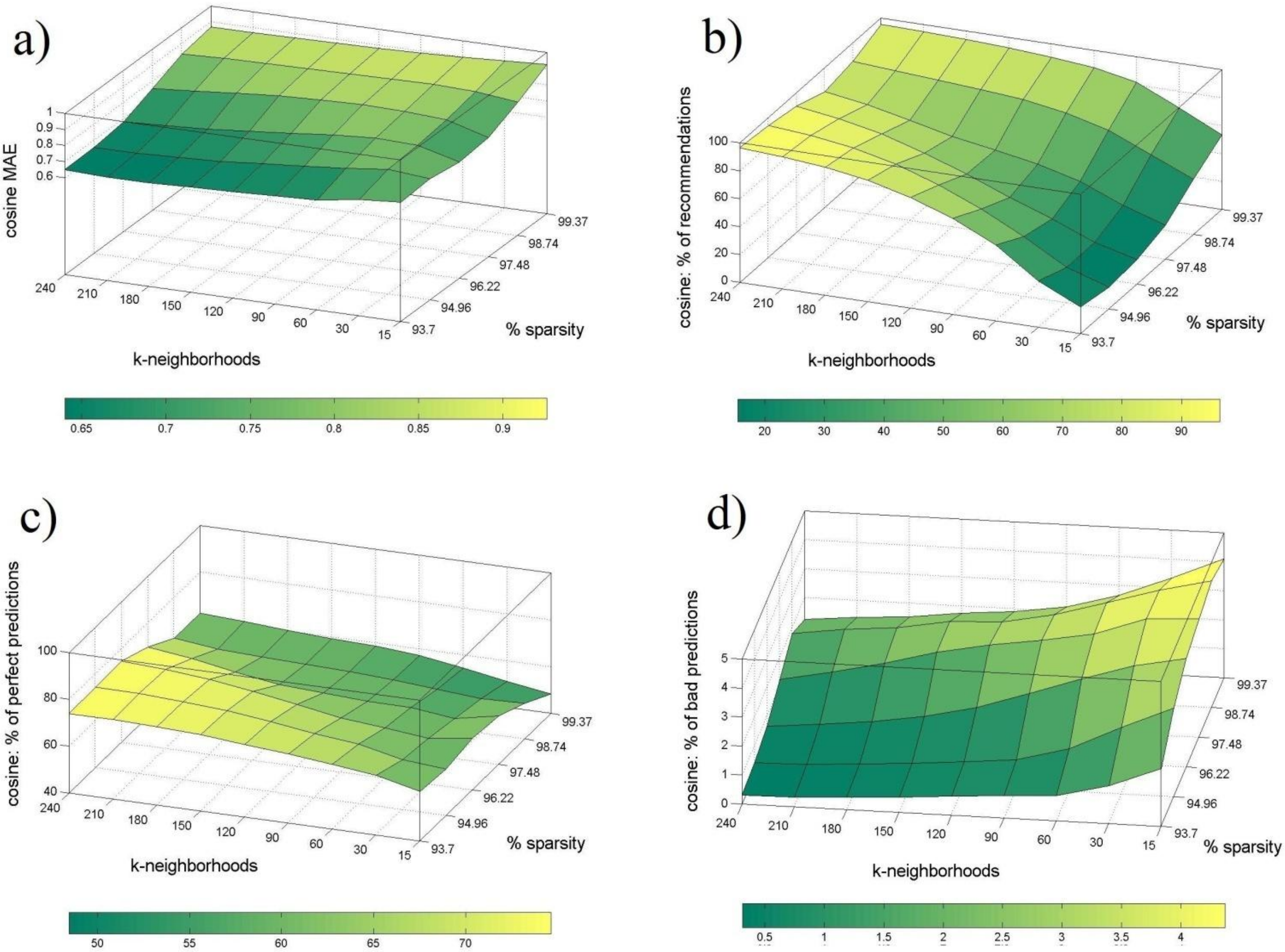

Figure 6. Results of the cosine metric: a) Mean Absolute Error, b) percentage of recommendations made, c) percentage of perfect predictions, d) percentage of bad predictions

5 Conclusions

The sparsity levels of RS databases have an important influence on the results of content-based collaborative filtering metrics. The impact of the sparsity influence depends on the k-neighborhood value selected, the main objective we want to maximize (MAE, capacity to recommend, etc) and, logically, on the metric used.

When the sparsity level increases:

- The Pearson correlation metric improves its MAE and has a negative effect on the capacity to recommend. In addition, its percentage of good predictions shows a slight increase.
- The cosine metric has a negative effect on all the aspects studied (MAE, capacity to recommend, correct predictions, incorrect predictions); this negative behavior can be reduced by selecting high k-neighborhood values.
- The Mean Squared Difference (MSD) greatly improves all the results except for the capacity to recommend.

The correlation metric obtains much better results than the cosine metric when working with sparse RS databases, especially when the k-neighborhood value is not high (preferably 60 and 90).

By using databases with a high degree of sparsity, the MSD metric obtains better results than the correlation metric in all the aspects studied except for the capacity to generate a large number of predictions.

The MSD metric presents unusually good behavior when applied to sparse RS ratio matrices. However, it should be used with caution due its very poor capacity to generate recommendations and its high probability of suffering from the effects of overspecialization; nevertheless, the MSD metric offers a serious alternative to the standard metrics when it is used in sparse ratio matrices and can be selected as a reference in designing new content-based CF metrics capable of satisfactorily tackling the RS sparsity problem.

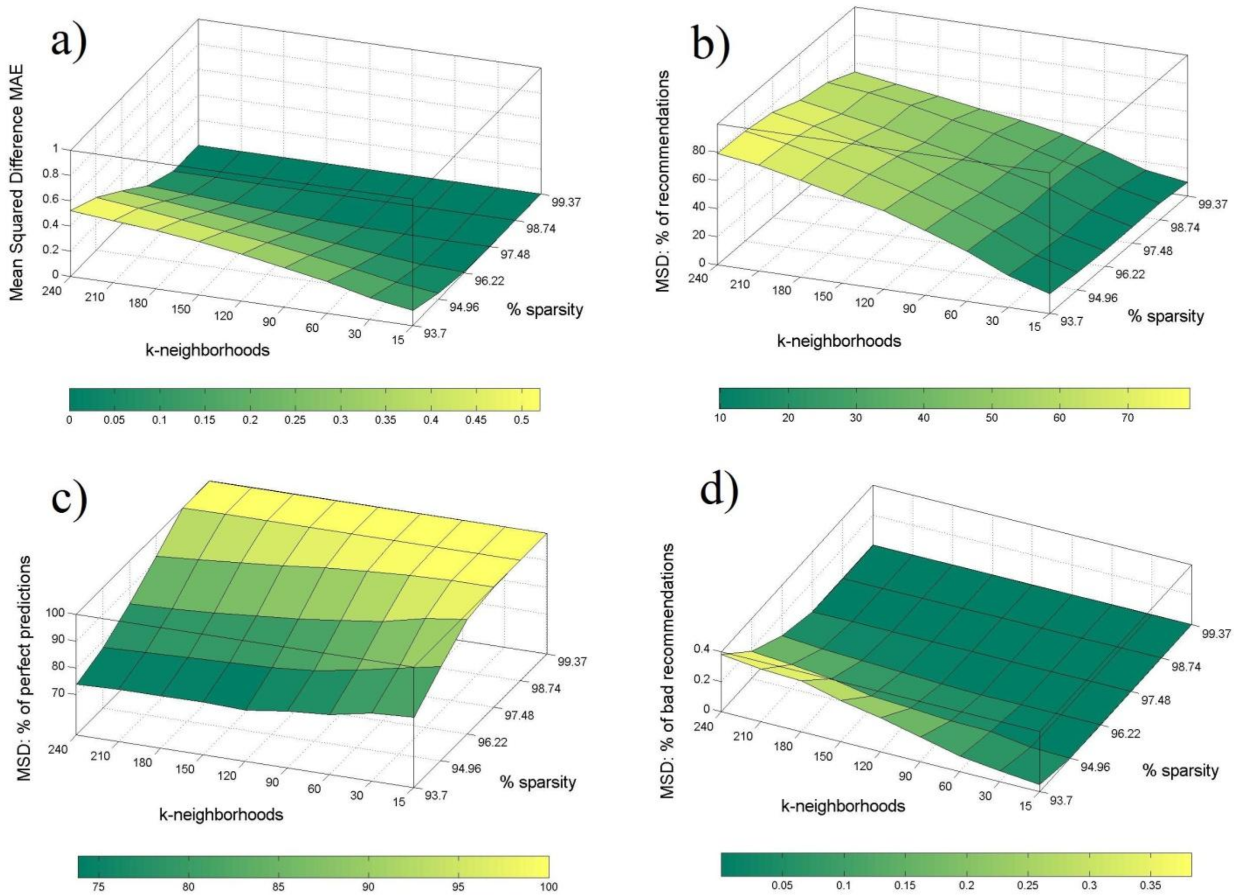

Figure 7. Results of the MSD metric: a) Mean Absolute Error, b) percentage of recommendations made, c) percentage of perfect predictions, d) percentage of bad predictions

6 References

1. Knights, M. Web 2.0, IET Communications Engineer, (February-March 2007), 30-35
2. Lin, K.J., Building Web 2.0, Computer, (May 2007), 101-102
3. Janner, T., Schroth, C. Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services, IT Pro, (May-June 2007), 36-41
4. Jinghua, H., Kangning, W., Shaohong, F. "A Survey of E-Commerce Recommender Systems", in Proceedings of the International Conference on Service Systems and Service Management, 2007, 10.1109/ICSSSM.2007.4280214, 1-5
5. Baraglia, R., Silvestri, F. "An Online Recommender System for Large Web Sites", in Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, 2004, 10.1109/WI.2004.10158, 199-205
6. Fesenmaier, D.R., Gretzel, U., Knoblock, C., Paris, C., Ricci, F., Stabb, S., Werther, H., Zipf, A. Intelligent Systems for Tourism, Intelligent Systems, vol. 17, no. 6, (nov/dec 2002), 53-66
7. Konstan, J.A., Miller, B.N., Riedl, J. PocketLens: Toward a Personal Recommender System, ACM Transactions on Information Systems, vol. 22, no. 3, (July 2004), 437-476.
8. Antonopoulus, N., Salter, J., CinemaScreen Recommender Agent: Combining Collaborative and Content-Based Filtering, IEEE Intelligent Systems, (January/February 2006), 35-41
9. Li, P., Yamada, S. "A movie recommender system based on inductive learning" in Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, vol. 1, 318-323
10. Adomavicius, Tuzhilin, A. Toward the Next Generation of Recommender Systems: a survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Enginnering, vol. 17, no. 6, (June 2005), 734-749
11. Herlocker, J. L., Konstan, J.A., Riedl, J.T., Terveen, L.G. Evaluating Collaborative Filtering Recommender Systems, ACM Transactions on Information Systems, vol. 22, no. 1, (January 2004), 5-53
12. Breese, J.S., Heckerman, D., Kadie, C. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering", in Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1998, 43-52
13. Kong, F., Sun, X., Ye, S. A Comparison of Several Algorithms for Collaborative Filtering in Startup Stage, In Proceedings of the IEEE networking, sensing and control, (March 2005), 25-28

14. Cho, S.B., Hong, J.H., Park, M.H. Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices, Lecture Notes on Computer Science, vol. 4611, (August 2007), 1130-1139

15. Ingoo, H., Kyong, J.O., Tae, H.R. The Collaborative Filtering Recommendation Based on SOM Cluster-Indexing CBR, Expert Systems with Applications, vol. 25, 2003, 413-423

16. Yager, R.R. Fuzzy Logic Methods in Recommender Systems, Fuzzy Sets and Systems, vol. 136, no. 2, (June 2003), 133-149

17. Linden, G., Smith, B., York, J., Amazon.com Recommendations, IEEE Internet Computing, (January-February 2003), 76-80

18. Giaglis, G.M., Lekakos, Improving the Prediction Accuracy of Recommendation Algorithms: Approaches Anchored on Human Factors, Interacting with Computers, vol. 18, no. 3, 2006, 410-431

19. Li, Y., Nayak, R., Weng, L.T., Xu, Y., "An Improvement to Collaborative Filtering for Recommender Systems", in Proceedings of the IEEE International Conference on Computational Intelligence for Modelling, Control and Automatitation, 2005

20. Fuyuki, I., Quan, T.K., Shinichi, H., "Improving Accuracy of Recommender Systems by Clustering Items Based on Stability of User Similarity", in Proceedings of the IEEE International Conference on Intelligent Agents, Web Technologies and Internet Commerce, 2006

21. Manolopoulus, Y., Nanopoulus A., Papadopoulus A.N., Symeonidis P. Collaborative Recommender Systems: Combining Effectiveness and Efficiency, Expert Systems with Applications, 2007. in press.

22. Herlocker, J., Konstan J., Riedl, J. An empirical Analysis of Design Choices in Neighborhood-based Collaborative Filtering Algorithms, Information Retrieval, vol.5, no. 4, , 2002, 287–310.

23. Pazzani, M. A Framework for Collaborative, Content-Based and Demographic Filtering, Artificial Intelligence Rev., December 1999, 393-408.

24. Sarwar, B., Karypis, G., Konstan J., Riedl, J. Application of Dimensionality Reduction in Recommender Systems-A Case Study, in Proceedings of ACM WebKDD Workshop, 2000.

25. Goldbergh, K., Roeder, T., Gupta D., Perkins D., Eigentaste: A Constant Time Collaborative Filtering Algorithm, Information Retrieval, Vol. 4, 2001, 133-151..

26. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer T.K., Harshman, R., Indexing by Latent Semantic Analysis, in Proceedings of the JASIS, Vol 41(6), 1990.

27. Hofmann, T., Collaborative Filtering via Gaussian Probabilistic Latent Semantic Analysis, in Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval, 2003.

28. Papagelis, M., Plexousakis D. Kutsuras, T., Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences, Lectures Notes on Computer Science, Vol. 3477, 2005, 224-239.

29. Bruyn, A.D., Giles C.L., Pennock, D.M., Offering Collaborative-Like Recommendations when Data is Sparse: the Case of Attraction-Weighted Information Filtering, in Proceedings of Lecture Notes in Computer Science, Vol. 3137, 2004, 393-396.

30. Polcicova G., Peter Tino, Making Sense of Sparse Rating Data in Collaborative Filtering via Topographic Organization of User Preference Patterns, Neural Networks, Vol. 17, 2004, 1183-1199.

31. http://www.movielens.org.