Using Neighborhood Pre-computation to Increase Recommendation Efficiency

Vreixo Formoso, Diego Fernández, Fidel Cacheda and Víctor Carneiro
University of A Coruña, Campus de Elviña s/n, 15017 A Coruña, Spain

Keywords: Collaborative Filtering, Nearest Neighbors, Performance.

Abstract: Collaborative filtering is a very popular recommendation technique. Among the different approaches, the *k*-Nearest Neighbors algorithm stands out by its simplicity, and its good and explainable results. This algorithm bases its recommendations to a given user on the opinions of similar users. Thus, selecting those similar users is an important step in the recommendation, known as neighborhood selection. In real applications with millions of users and items, this step can be a serious performance bottleneck because of the huge number of operations needed. In this paper we study the possibility of pre-computing the neighbors in an offline step, in order to increase recommendation efficiency. We show how neighborhood pre-computation reduces the recommendation time by two orders of magnitude without a significant impact in recommendation precision.

1 INTRODUCTION

Recommender systems are a popular technique in fields such as e-commerce, where they help users to find the products they need. A particularly successful technique is collaborative filtering, that computes high-quality recommendations to a user, based on the opinions of other users with similar tastes or interests.

Among the different algorithms developed, the *k*-Nearest Neighbors (kNN) approach is very popular because it is simple, intuitive (which allows to justify a recommendation decision), and does not require a training step (Desrosiers and Karypis, 2011). It first selects a set of neighbors, that is, the set of users most similar to the user the system is generating a recommendation to (known as the active user). The items most highly rated by those neighbors are the ones recommended.

The neighborhood selection is a computationally intensive step, that can take a long time. It requires to compute the similarity between the active user and every remaining user. Each similarity computation also requires to compare the opinions or ratings of both users. In real applications, with millions of users and items, this can take several seconds, which is unacceptable in many cases where the recommendations need to be generated in real time. Although techniques such as compression (Cöster and Svensson, 2002) have been proposed to optimize it, neighborhood computation remains a very expensive step.

A practical solution is to pre-compute the neighborhood in an offline step, storing it in an index-like structure for later usage. This can significantly reduce the recommendation time. However, the neighborhood should be updated to include new user opinions. Otherwise, the neighbors used for recommendation may not reflect the current user tastes, and that might negatively influence the quality of the recommendations.

In this paper we study the impact of neighborhood pre-computation, both on computational efficiency and recommendation quality. We show how it is a very effective technique to reduce recommendation time, without significantly reducing the recommendation quality.

2 EXPERIMENTS AND RESULTS

For our experiment we have used the Netflix dataset (Bennett and Lanning, 2007), a popular dataset from the movie recommendation domain. It contains over 100 million ratings from 480,189 users to 17,770 movies, collected between October 1998 and December 2005. In order to evaluate the impact of neighborhood pre-computation, we have studied the evolution of the precision of the results with the time elapsed after pre-computation.

From all ratings available in the dataset, we have taken those ratings given before January 1st, 2005,

Formoso V., Fernández D., Cacheda F. and Carneiro V..
Using Neighborhood Pre-computation to Increase Recommendation Efficiency.
DOI: 10.5220/0004139703330335
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2012), pages 333-335
ISBN: 978-989-8565-29-7
Copyright © 2012 SCITEPRESS (Science and Technology Publications, Lda.)

and we have pre-computed the neighborhood taking only those ratings into account. Then, we evaluate the algorithm considering all ratings up to February 1st, March 1st, and so on. For the evaluation, we have considered two situations: one with neighborhood pre-computation, where the previously computed neighborhood (with ratings up to January 1st) is used, and another where the neighborhood is computed at recommendation time (and thus all ratings up to the given month are considered). In either case, for recommendation we consider all the ratings available at that time. This way, we simulate an environment where the neighborhood is computed once at the beginning of the year, but the rating matrix is being updated constantly. We have performed the evaluation with $1,000$ randomly selected users.

First, we have studied how neighborhood pre-computation can speed up recommendation time. For our experiments, we have used a PC with a Intel Pentium 4 CPU at 3.20 GHz and 256 MiB of RAM. Using an old machine is an approach commonly used for efficiency evaluation in Information Retrieval (Badue et al., 2007) when the dataset used is significantly smaller than the amount of data in real applications.

Results are shown in Figure 1. As expected, the usage of neighborhood pre-computation significantly reduces recommendation time. In average, recommendation is computed two orders of magnitude faster, which is a very important achievement. Moreover, with neighborhood pre-computation the required time remains more or less constant among months, even though the number of ratings increases. On the other hand, with no pre-computation it significantly increases with the number of ratings. That is, the neighborhood computation time is more affected by the number of ratings than the final recommendation step, which makes sense because in that final step only a few users (the neighbors) are actually considered.

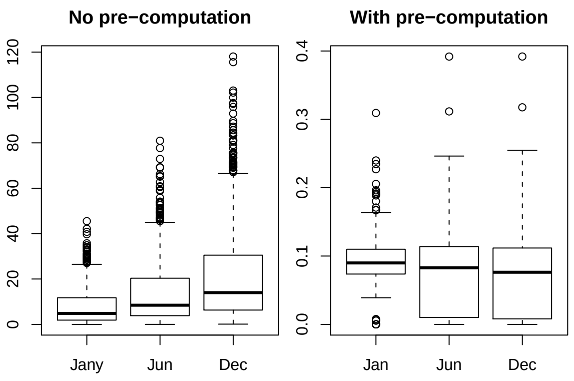

Figure 1: Recommendation time (seconds) with and without neighborhood pre-computation. Note the different scale in each chart.

We have also evaluated the precision and recall of the recommendations, in order to study the evolution of the quality with the time elapsed after neighborhood pre-computation. If the precision dropped very fast, this technique would be not very useful, because the pre-computation step would need to be done very often. However, as seen in Table 1, this is not the case. Both precision and recall remain similar with and without pre-computation[1], without statistical significant differences between them. While updating the rating matrix is very important (in order to recommend new products, for example), dealing with an old neighborhood seems to have almost no impact in recommendation quality. Of course, the actual threshold where an outdated neighborhood begins to negatively impact quality is domain-dependent. While a several months old neighborhood is not a problem in the studied case, other domains might require a shorter neighborhood update time.

Table 1: Precision@5 and Recall@5 with and without pre-computation.

	P@5		R@5	
	With	Without	With	Without
Jan	1.28	1.28	0.13	0.13
Feb	0.99	0.90	0.12	0.08
Mar	1.16	1.39	0.13	0.15
Apr	0.98	1.21	0.12	0.17
May	0.72	0.76	0.07	0.08
Jun	0.75	0.81	0.09	0.12
Jul	1.03	0.65	0.46	0.12
Ago	0.12	0.39	0.03	0.05
Sep	0.26	0.32	0.08	0.27
Oct	0.22	0.21	0.14	0.12

3 CONCLUSIONS

In this paper we have evaluated the benefits of neighborhood pre-computation. We have shown how this technique can reduce the recommendation time of *k*-Nearest Neighbors algorithms by two orders of magnitude, without a significant impact in the recommendation list quality. These results show that real applications can benefit from neighborhood pre-computation techniques with no important drawback in terms of precision. In the future, we plan to extend this research to further domains. We also plan to study the impact on different metrics, and with different update strategies.

[1]Note that bad precision in the last months is related to the evaluation methodology, as there are few relevant ratings after that time. This is a well-known limitation of offline evaluation (Cacheda et al., 2011).

ACKNOWLEDGEMENTS

This research was supported by the Ministry of Education and Science of Spain and FEDER funds of the European Union (Project TIN2009-14203).

REFERENCES

Badue, C. S., Baeza-Yates, R., Ribeiro-Neto, B., Ziviani, A., and Ziviani, N. (2007). Analyzing imbalance among homogeneous index servers in a web search system. *Inf. Process. Manage.*, 43:592–608.

Bennett, J. and Lanning, S. (2007). The netflix prize. In *KDDCup '07: Proceedings of KDD Cup and Workshop*, page 4, San Jose, California, USA. ACM.

Cacheda, F., Carneiro, V., Fernández, D., and Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Trans. Web*, 5:2:1–2:33.

Cöster, R. and Svensson, M. (2002). Inverted file search algorithms for collaborative filtering. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 246–252, New York, NY, USA. ACM.

Desrosiers, C. and Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 107–144. Springer.