Simple Time-Biased KNN-based recommendations

Pedro G. Campos[1,2], Alejandro Bellogín[1], Fernando Díez[1], J. Enrique Chavarriaga[1]

[1]Universidad Autónoma de Madrid
Francisco Tomás y Valiente 11
28049 Madrid, Spain
+34 91 4972213

[2]Universidad del Bío-Bío
Av. Collao 1202
4081112 Concepción, Chile
+56 41 2731517

{pedro.campos, jes.chavarriag}@estudiante.uam.es
{alejandro.bellogin, fernando.diez}@uam.es

ABSTRACT

In this paper, we describe the experiments conducted by the *Information Retrieval Group* at the Universidad Autónoma de Madrid (Spain) in order to better recommend movies for the 2010 CAMRa Challenge edition. Experiments were carried out on the dataset corresponding to *weekly Filmtipset* track. We consider simple strategies for taking into account the temporal context for movie recommendations, mainly based on variations of the KNN algorithm, which has been deeply studied in the literature, and one ad-hoc strategy, taking advantage of particular information in the weekly Filmtipset track. Results show that the usage of information near to the recommendation date alone can help improving recommendation results, with the additional benefit of reducing the information overload of the recommender engine. Furthermore, the use of social interaction information shows also a contribution in order to better predict a part of users' tastes.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – Information Filtering, Retrieval Models, Selection Process; I.5.1 [Pattern Recognition] - Models

General Terms

Algorithms, Performance.

Keywords

Recommender Systems, Movie Recommendation, Temporal Information.

1. INTRODUCTION

Time has become an important dimension to analyze in Recommender Systems (RS). Although until recently this aspect was not investigated thoroughly within RS scope, recent work show that taking into account this dimension can improve accuracy of recommendations [4]. Many ideas to handle temporal information have been proposed. In this work we discuss some of them, and apply a few on the Weekly Filmtipset dataset of 2010 CAMRa Challenge [7], presenting the most relevant results. The remainder of the paper is structured as follows: section 2 presents related work. Section 3 describes experiments performed. Section 4 discusses the results obtained. Finally, section 5 presents some preliminary conclusions and expected future work.

CAMRa 2010, September 30, 2010, Barcelona, Spain.

2. RELATED WORK

One of the first ideas for incorporating temporal information in RS was that of incrementing the weight of recent ratings [3], assuming that the most recent preferences of a user reflects in a better way his/her actual preferences (and near-future ones). In this work, a neighborhood of items is determined, and the final rating prediction is weighted according to the difference between the rating of each user in the neighborhood of the active user, and the most recent rating of the active user for an item in the neighborhood of the objective item. The work of Tang et al. [8] can be considered a special case of this idea, where the rating data of older items was truncated, leaving only the most recent items as input for the recommendation engine. This was a movie RS, and the production year of the movies was used to decide if the movie should be eliminated from the database or not.

A simple idea that shows improvements on recommendations' accuracy is the one proposed by Lee et al. [6], where two temporal dimensions are considered, the time that an item has been included in the RS, and the time that a user showed a preference for the item, in an increasing weight scheme depending on the information recency. An interesting point in this work was the usage of implicit information. Another idea that has been explored is the usage of different K values in K-Nearest Neighbor (KNN) algorithms, as a time function (more specifically, K values that minimize the error in different time intervals are searched) [5]. The use of matrix factorization techniques has also been extended in order to incorporate temporal information. In particular, several factors in the factors model that represent long and short-term changes in users' behavior and items acceptance [4].

3. EXPERIMENTS

This section details how we performed our experiments, including basic descriptions of the implemented strategies for generating recommendations. For each strategy, we generate a list of recommended movies (with the predicted rating) for each user in test set, and then calculate the results metrics detailed in section 4.

3.1 Data Pre-processing

The basic strategies carried out only used as input data the movie ratings available in the datasets. In order to make use of all the information (using only the allowed information for each task), we created two versions of ratings datasets, one for the Christmas week (XmasFullDataset) containing ratings in ratings_train.tsv before December 21th, 2009, and other for the Oscar week (OscarFullDataset), containing all ratings in ratings_train.tsv and ratings_test_xmas before February 27th, 2010,

3.2 Baseline strategy

In order to have a baseline to compare results of our strategies, we use a trivial user based KNN algorithm separately over both datasets, with K=3. Similarities between users are calculated using the common Pearson Correlation and predictions are calculated as aggregations of the ratings of the most similar users, as described in [1]. In order to obtain recommendation lists, we calculate predictions for all movies for each user, selecting the top N with respect to the predicted rating.

3.3 Simple KNN Strategy

In order to obtain more results, the first strategy was to vary the K value for the KNN algorithm. We use values of K from 1 up to 50.

3.4 Ad-hoc Strategy

As a second strategy we created an ad-hoc recommender using specific information provided for the Challenge. In this case, we use data from reviews (in reviews.tsv file), and movie comments (in moviecomments.tsv file) as they include time stamps. This strategy considers that social interaction influence users actions and tastes. We estimate the preference of a user towards an item taking into account which movies the user has reviewed and how similar are each of these movies with respect to the objective movie (using Pearson correlation). Besides, we also take into account the portion of time between the review was made and the recommendation date.

3.5 Time-Biased KNN

The third strategy is a simplification of the increasing weight in function of time scheme. In this case, we calculate the most similar users in datasets with all the available information (XmasFullDataset and OscarFullDataset), but after that we use only the most recent ratings of the neighbors to estimate the prediction of the rating of the active user, assuming that recent ratings corresponds to the actual preferences of the users (and that similar users tend to be similar along time). This can be summarized in eq. (1).

$$\hat{r}(x,s) = \bar{r}_x + \frac{\sum_{x' \in N[x]} sim(x,x') \times \left(rr_{x',s} - \overline{rr}_{x'}\right)}{\sum_{x' \in N[x]} |sim(x,x')|} \qquad (1)$$

Where $\hat{r}$ is the prediction for the rating, $\bar{r}_x$ is the mean rating of user x, $sim(x,y)$ is the similarity between user x and user y, $N[x]$ is the set of nearest neighbors of x, rr is a recent rating from a user to an item and $\overline{rr}$ stands for the mean recent rating of a user. A disadvantage of this strategy is that if the time interval is too small, then there could be not enough information as to make a prediction. Within this scheme (and remaining Time-Periodic Biased KNN strategy), in cases where no data was available to calculate the neighbors of a user, the average rating of the movie or of the user was used instead, in that order. We tested this strategy for varying values of K, with 2 datasets of recent ratings for each task. In the case of both task, the recent ratings datasets were 1 month and 4 months (which includes all data in 1 or 4 months previous to the starting day of recommendation). The selection of these short time horizons responds to the premise that many movie preferences remain only for a short time-span. The use of wider horizons within a weight-decay scheme was not considered because of the Challenge deadline constraints.

3.6 Time-Periodic-Biased KNN (TPB KNN)

This strategy is a variation of the Time-Biased KNN strategy, in which the recent ratings datasets includes data from the last months immediately before the recommendation weeks, and also data for the same months and days, but in the previous year. In the case of both tasks, the recent ratings datasets for this strategy were: 1 month and 2 months.

4. RESULTS

For each strategy, we calculate MAP, P@5, P@10, AUC and NDCG. The metric values are calculated with the trec_eval utility[1], a public program to evaluate TREC results using the standard NIST evaluation procedures. Within this scheme, each user is treated as a query, and the recommendation list is treated as the results for the query. This way, using the test sets as ground truth, trec_eval is able to calculate MAP, P@5, P@10 and NDCG as usual in Information Retrieval. We use AUCCalculator utility [2] to calculate AUC. We have included NDCG mainly because, in terms of Information Retrieval, if we consider users as queries, recommended items as documents resulting from the query, and the predicted ratings as approximations to the scores given by the search engine, we can compute the cumulative gain (CG) at position p of a particular rating. Each user (i.e., a query) has a discounted CG. If we normalize it using the information of the whole set of users we compute the NDCG measure, which allows fair comparisons between different algorithms. Besides that, it helped us to decide which algorithm and which parameter combination (among all the combinations tried) performed better in case of equal performances using other measures.

The results obtained for the above-mentioned metrics with the baseline strategy (see section 3.2) are shown in Table 1.

Table 1. Baseline strategy results.

Task	MAP	P@5	P@10	AUC	NDCG
Christmas	0,0025	0,0051	0,0039	0,0265	0,0059
Oscar	0,0021	0,0075	0,0051	0,0345	0,0064

As we can see here, the results are extremely poor with this strategy. In terms of Precision, AUC and NDCG, these results show that task 1 (Christmas week) is somewhat more difficult than task 2 (Oscar week). The simple KNN strategy with all ratings (KNN at different K values) is the first improving strategy that we applied. Figure 1 and Figure 2 show the performance for each metric in task 1 and task 2 respectively. The best values for the considered metrics are detailed in Table 2.

Table 2. Best results for simple KNN with averages strategy.

Task	K value	MAP	P@5	P@10	AUC	NDCG
Christmas	K=4	0,0030	0,0063	0,0042	0,0311	0,0076
	K=20	0,0047	0,0057	0,0056	0,0893	0,0250
	K=49	0,0062	0,0062	0,0053	0,1453	0,0414
	K=50	0,0060	0,0058	0,0053	0,1461	0,0416
Oscar	K=15	0,0044	0,0091	0,0074	0,0897	0,0227
	K=42	0,0053	0,0076	0,0078	0,1467	0,0409
	K=50	0,0056	0,0076	0,0072	0,1593	0,0443

[1] http://trec.nist.gov/trec_eval/

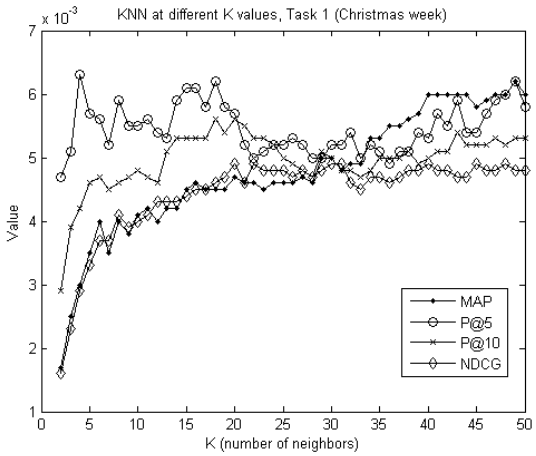

Figure 1. Results for task 1 with simple KNN strategy at different K values.

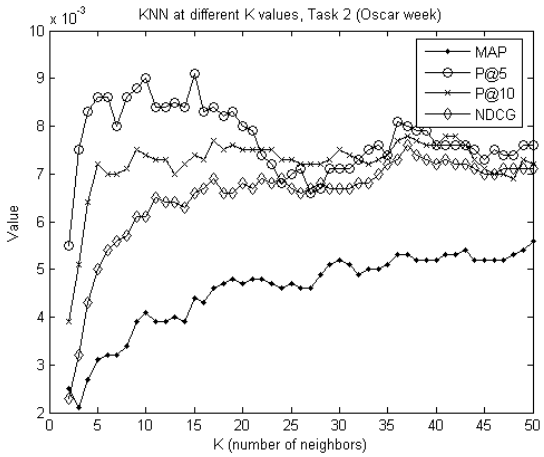

Figure 2. Results for task 2 with simple KNN strategy at different K values.

As we can see in Table 2, there is no optimal K value that performs the best for all metrics. However, from the precision standpoint, a K value in the range [15 – 20] seems to work right, meanwhile MAP shows better values as K increases (see figures 1 and 2).

In Table 3 we show the results for the ad-hoc strategy. The "review week" parameter represents how many weeks are considered until the last allowed date for each task. In this way, if this value is negative (e.g. Week -*n*), we take into account *n* weeks into the past. We also include results found when the reviews are from the same (Week 0) or the next (Week +1) week of the evaluation. We are aware this scenario is not real, but it gives us a hint about which could be the best achievable value using this approach.

It is interesting to note that, for task 1, the best MAP and precision values are obtained with data from the week previous to the recommendation, and not with data from the same week (which does occur on task 2). However, in terms of NDCG, best results are always found when using the evaluation week on its own.

These results have a very important output: social interaction between users and movies (reviews, comments, etc.) must be taken into account in a social recommender website, since it, probably, will affect subsequent user actions, such as purchases or ratings. It follows from the fact that values over 0 on the metrics indicate that users actually see (and like) movies that are similar to previously reviewed movies. However, we need to improve this scheme, as metric values are very low (even worse than baseline results), which may indicate that movies recommended with this strategy are not high rated (we are not aware if the reviews and comments are positive or negative). On the other hand, this recommender can be implemented in a scalable and incremental way, and, besides that, it can provide with straightforward recommendation explanations, which is still an open problem in RS [1].

Table 3. Results of ad-hoc strategy.

Task	Review Weeks	MAP	P@5	P@10	AUC	NDCG
Christmas	Week +1	0,0025	0,0008	0,0012	0,2087	0,1539
	Week 0	0,0025	0,0008	0,0012	0,2087	0,1539
	Week -1	0,0059	0,0018	0,0023	0,2559	0,0791
	Week -2	0,0041	0,0013	0,0012	0,2494	0,0739
	Week -3	0,0036	0,0011	0,0009	0,2465	0,0723
	Week -4	0,0035	0,0008	0,0008	0,2449	0,0716
Oscar	Week +1	0,0036	0,0022	0,0018	0,2514	0,0662
	Week 0	0,0028	0,0021	0,0018	0,2102	0,1573
	Week -1	0,0015	0,0003	0,0005	0,1540	0,0296
	Week -2	0,0012	0,0007	0,0005	0,2039	0,0414
	Week -3	0,0012	0,0007	0,0005	0,2039	0,0414
	Week -4	0,0021	0,0004	0,0004	0,2364	0,0610

Figure 3 shows results of the Time-Biased KNN strategy, with the 1 month recent ratings dataset (see section 3.5), for task 1. Similar results were seen on task 2. This strategy was tested for K values in the range [2 - 50]. Table 4 shows the best results for different configurations of this strategy on the Christmas and Oscar task.

In this case, results are much better than those obtained with the previous strategies, particularly on the task 1 (Christmas week). Figure 3 shows that MAP does not continue increasing its value as K increases when using this strategy (on the opposite to the simple KNN strategy). In the case of the 2nd task, the results were quite similar to these ones.

Table 4. Best results for Time-Biased KNN strategy.

Task	Time Interval	K value	MAP	P@5	P@10	AUC	NDCG
Christmas	1 Month	K=14	0,0405	0,0070	0,0044	0,4552	0,3890
		K=25	0,0399	0,0018	0,0009	0,4515	0,3891
	4 Months	K=11	0,0339	0,0018	0,0018	0,4415	0,3768
		K=24	0,0341	0,0018	0,0009	0,4412	0,3770
Oscar	1 Month	K=22	0,0381	0,0033	0,0022	0,4226	0,3770
		K=23	0,0381	0,0033	0,0017	0,4229	0,3777
	4 Months	K=21	0,0359	0,0034	0,0028	0,4161	0,3753
		K=25	0,0360	0,0022	0,0022	0,4163	0,3754

These results show that better predictions are obtained with recent data. So it confirms our intuition that recent ratings better reflects actual users' tastes. Table 5 shows results from the TPB KNN strategy. In this case, for the 1 month interval, values of K in the range of [2 - 50] were tested, but for the 2 months interval only K values in the range [2-16] were tested, due to time constraints.

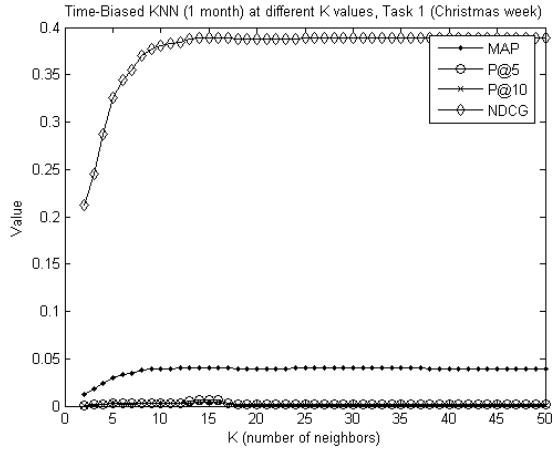

Figure 3. Results for task 1 with Time-Biased KNN strategy at different K values, using ratings from the last month.

Table 5. Best results for TPB KNN strategy.

Task	Time Interval	K value	MAP	P@5	P@10	AUC	NDCG
Christmas	1 Month	K=40	0,0326	0,0000	0,0000	0,4407	0,3755
	2 Months	K=13	0,0338	0,0000	0,0000	0,4450	0,3765
Oscar	1 Month	K=33	0,0344	0,0024	0,0018	0,4101	0,3785
	2 Months	K=13	0,0188	0,0019	0,0014	0,2733	0,3133

These results are also competitive, but not as good as those obtained with the Time-Biased KNN strategy, meaning that the additional information of the same period but on the past year is not contributing to better predictions of the actual users' tastes.

It is interesting to note that MAP and NDCG values are consistently better with the time-biased strategies; meanwhile P@N values are somewhat lower than the baseline and simple KNN strategies. This is probably due to a better capacity of the time-biased strategies to recommend movies highly rated by users. We must remember that in the calculation of the Precision measure, documents (items recommended) are only considered as relevant or non-relevant, whereas in NDCG the relevance value (rating) is taken into account. This way, if time-biased KNN strategies are able to recommend the same number or even less movies, but with higher rating, NDCG will increase, meanwhile Precision will maintain or even decrease (as actually happens).

5. CONCLUSIONS AND FUTURE WORK

The results of the different strategies presented in this paper show the difficulties for getting good predictions from the dataset. Although the metric values could be considered low, the time-biased strategies show considerable improvements compared with our baseline and ad-hoc strategy. This is a proof of the potential that temporal information can be an important input in getting better predictions of users' tastes. Further study on the characteristics of this specific dataset should help us to come with more accurate predictions. Due to time constraints we could not try some other schemes that probably can provide better predictions, such as the use of matrix factorization models, or the construction of a hybrid recommender which takes into account social and temporal information. Tests made with social information (reviews and comments) showed that social relationships does have an impact on users' actions (in this case ratings), and an adequate combination of all this information will surely lead to better recommendations.

6. ACKNOWLEDGMENTS

This research was supported by the Spanish Ministry of Science and Innovation (TIN2008-06566-C04-02) and the Scientific Computing Institute at UAM. The first author acknowledges support from the Chilean Government through the Becas-Chile scholarship program.

7. REFERENCES

[1] G. Adomavicius, A. Tuzhilin. Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering,17(6):734-749, June 2005.

[2] J. Davis, M. Goadrich. The Relationship Between Precision-Recall and ROC Curves. 23rd International Conference on Machine Learning (ICML), Pittsburgh, PA, USA, 26th - 28th June, 2006.

[3] Y. Ding, X. Li, M. E. Orlowska. Recency-based collaborative filtering. In ADC'06 Proceedings of the 17th Australasian Database Conference, pp. 99-107, Darlinghurst, Australia, Australia, Australian Computer Society, Inc., 2006.

[4] Y. Koren. Collaborative filtering with temporal dynamics. In KDD '09 Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 447-456, New York, NY, USA, ACM, 2009.

[5] N. Lathia, S. Hailes, L. Capra. Temporal collaborative filtering with adaptive neighbourhoods. In SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pp. 796–797, New York, NY, USA, ACM, 2009.

[6] T. Q. Lee, Y. Park, Y. T. Park. A time-based approach to effective recommender systems using implicit feedback. Expert Syst. Appl. 34(4):3055–3062, 2008.

[7] A. Said, S. Berkovsky, E. W. De Luca. Putting Things in context: Challenge on Context-Aware Movie Recommendation. In CAMRa2010: Proceedings of the RecSys '10 Challenge on Context-aware Movie Recommendation, Barcelona, Spain, ACM, 2010.

[8] T. Y. Tang, P. Winoto, K. C. C. Chan. On the temporal analysis for improved hybrid recommendations. In WI '03 Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence, pp. 214, Washington, DC, USA, IEEE Computer Society, 2003.