Temporal Collaborative Filtering With Adaptive Neighbourhoods

Neal Lathia
Dept. of Computer Science
University College London
London, WC1E 6BT, UK
n.lathia@cs.ucl.ac.uk

Stephen Hailes
Dept. of Computer Science
University College London
London, WC1E 6BT, UK
s.hailes@cs.ucl.ac.uk

Licia Capra
Dept. of Computer Science
University College London
London, WC1E 6BT, UK
l.capra@cs.ucl.ac.uk

ABSTRACT

Collaborative Filtering aims to predict user tastes, by minimising the mean error produced when predicting hidden user ratings. The aim of a *deployed* recommender system is to *iteratively* predict users' preferences over a dynamic, growing dataset, and system administrators are confronted with the problem of having to continuously tune the parameters calibrating their CF algorithm. In this work, we formalise CF as a time-dependent, iterative prediction problem. We then perform a temporal analysis of the Netflix dataset, and evaluate the temporal performance of two CF algorithms. We show that, due to the dynamic nature of the data, certain prediction methods that improve prediction accuracy on the Netflix probe set do not show similar improvements over a set of iterative train-test experiments with growing data. We then address the problem of parameter selection and update, and propose a method to automatically assign and update per-user neighbourhood sizes that (on the temporal scale) outperforms setting global parameters.

Categories and Subject Descriptors: H.3.3 Information Search and Retrieval: Information Filtering

General Terms: Algorithms

Keywords: Temporal Collaborative Filtering.

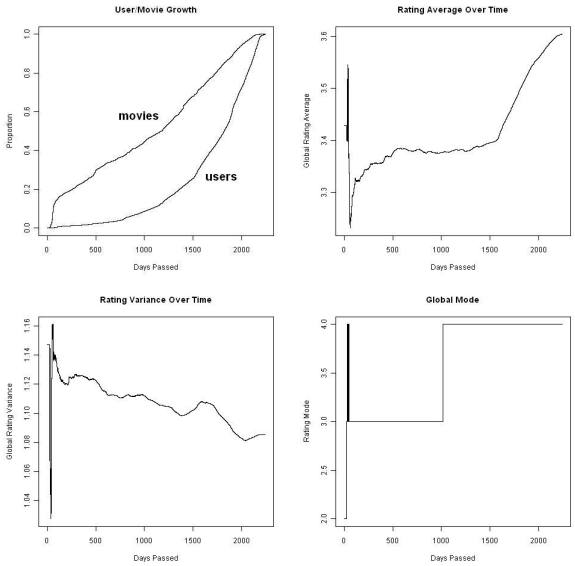

Figure 1: The Netflix Data Changes Over Time: Growth of Number of Movies & Users, and Changes in the Rating Mean, Variance, Mode

1. INTRODUCTION

Recommender Systems (RSs), based on Collaborative Filtering (CF), are becoming important portals via which users interact with online web sites. The core problem of CF is often reduced to one of *prediction*, and research in this field thus aims to optimise performance based on a mean error metric [1] - by testing on a partitioned dataset of user ratings. However, deployed implementations of these algorithms operate in a rather different setting. The *dataset* is dynamic: the user base and number of ratings grow over time, and the system will need to be iteratively updated. A growing dataset leads to changing *features* of the ratings; both global and user statistics derived from CF data (and often used to predict ratings) change over time. We illustrate many of the changes that the Netflix data[1] is subject to over time in Figure 1. The top-left plot shows the normalised near-linear growth of the number of movies over time, compared to the exponential growth of users. The remaining plots reflect how these changes influence the ratings: the global rating mean, variance, and mode vary as well. Consequently, static *parameters* may not always be optimal, and system administrators are required to continuously tune their algorithms for best performance. The aim of a recommender system is thus to continuously anticipate the behaviour of its users with a *changing* dataset; however, the algorithms applied in this context are not designed to adapt to meet the changes they will experience.

In this work, we extend CF evaluation to gain insight into the temporal performance of these algorithms as they are iteratively applied. We formalise CF as a time-dependent prediction problem and evaluate the *bias model* described by Potter [2] and the k-Nearest Neighbour algorithm [3] with a series of iterative experiments, to highlight the influence of changing data on prediction. Using this evaluation, we propose and evaluate *adaptive* temporal CF, a method of temporally adapting the size of user kNN neighbourhoods based on the performance measured up to the current time.

[1]http://www.netflixprize.com

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.
ACM 978-1-60558-483-6/09/07.

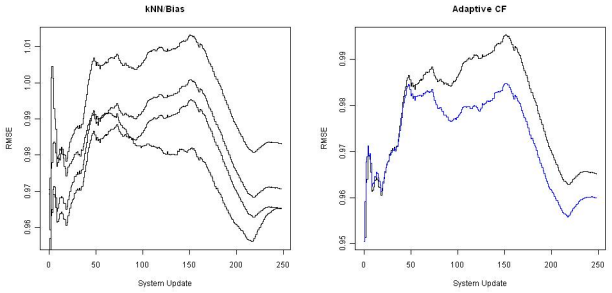

Figure 2: Time-Averaged RMSE for kNN & Bias Algorithm (left), Adaptive-kNN (right)

2. TEMPORAL EXPERIMENTS

We used 5 subsets of 60,000 users of the Netflix data. Starting at time $\epsilon = 500$ days from the first rating in the dataset, the RS is updated at every $\mu = 7$ days. At each time t, based on all ratings input prior to t, we aim to predict any ratings that will be input before time $(t + \mu)$. Time t is then incremented by μ and the process is repeated; our data allows for 250 temporal updates. This set up implies that, due to the temporal structure of the data, both the number of historical ratings and ratings predicted in $(t+\mu)$ will grow as t increases. We pruned each test set of any user-item pairs that have a history less than $h = 1$ (whether it be that the user has rated or the movie has been rated fewer than h times) in order to avoid testing extreme cold-start behaviour. This way, we aim to test CF algorithms on a wide range of dataset sizes and highlight the temporal performance of these algorithms.

We visualise the results using a time-averaged RMSE metric, or simply the RMSE achieved on all predictions made up to t. We plot the kNN and bias model cross-validated time-averaged RMSE results in Figure 2 (left). On the right hand part of the plot, the results (in descending order) are $k = 20, 35, 50$, and the bias model, which achieves the lowest RMSE. However, on the left hand side of the plot, the ordering of models is not the same: viewing the temporal results emphasises the difficulty of identifying one algorithm that *consistently* outperforms all others. For example, the bias model is the most accurate in two-thirds of the updates, while in the other cases the kNN model is more accurate. Temporal performance itself varies: after the $50th$ update predictive accuracy wanes, highlighting the dependence that these methods have on the *quality* of the data they train on. From these results, $k = 50$ emerges as the most temporally accurate kNN parameter. However, we also explored how prediction error is distributed across a community of *individuals* by splitting users into groups according to profile size and plotting the group's 5-fold cross-validated time-averaged performance. The k value performance in the group with fewer than 10 ratings is the *opposite* of what we observed when all groups were merged: larger neighbourhoods leads to *less* accurate results. In fact, there is often no consensus between the method that produces the best *global* performance and that which best suits *each user*.

2.1 Adaptive Neighbourhoods

The above experiments highlight how CF does not modify its behaviour based on how well it is performing over time, and, in particular, no single model will consistently outperfom all others. However, switching *between* models may add intractable complexity to the scalability of CF, as it would require multiple independent models to be run on the data at each update. In this work, we therefore experiment with temporal-adaptivity that sets the k parameter for the next system update based on selecting, for each user, the parameter that would have most improved current performance. We select k from subset of potential values $P = \{0, 20, 35, 50\}$. A $k = 0$ value disregards neighbourhoods completely; in this case we can either return a baseline item (b_i) or user (b_u) mean rating. We then proceed to set a value $k_{u,t} \in P$ for each user u at time t. When new users enter the system, their $k_{u,t}$ value is bootstrapped to a pre-determined member of P. At each time step t, all users u have a corresponding error value $e_{u,t}$ denoting the time-averaged RMSE achieved on the predictions made to date on *their profile*. The idea is for each $k_{u,(t+\mu)}$ to be set to that which would have provided the steepest improvement on the users' $e_{u,t}$ value in the last time step. We therefore aim to optimise the per-user k value by selecting the parameter that would have maximised the improvement on the current error:

$$\forall u : k_{u,t+1} = \max_{k_i \in P} (e_i - RMSE_{t,P_i}) \quad (1)$$

The time-averaged RMSE results are shown in the rightmost plot of Figure 2, compared to the results of the best *global* parameter setting ($k = 50$). The results highlight a number of benefits of adaptive CF. The adaptive strategy at first rivals the performance of $k = 50$, but then improve the overall time-averaged RMSE without requiring any manual parameter tuning. In particular, the user-adaptivity component of the strategy shows its effect since the results approximate or improves over simply selecting the best overall parameter. The improved accuracy of adaptive-kNN comes at little cost: computing predictions remains the same, since, for example, the computations for both the $k = 20$ and 35 predictions for a user-item pair are contained within those required to compute with $k = 50$.

3. CONCLUSION

This work departs from traditional CF research by extending the analysis of prediction performance to incorporate a sequence of classification iterations that learn from a growing set of ratings. We then implemented and evaluated a cheap method that automatically tunes parameters to provide greater temporal accuracy. Due to limited space, we point the reader to [4] for full details[2], results, and discussion of this work.

4. REFERENCES

[1] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM TOIS*, 22(1):5–53, 2004.

[2] G. Potter. Putting the Collaborator Back Into Collaborative Filtering. In *Proceedings of the* 2^{nd} *Netflix-KDD Workshop*, 2008.

[3] R. Bell and Y. Koren. Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights. In *IEEE ICDM*. IEEE, 2007.

[4] N. Lathia, S. Hailes, and L. Capra. Temporal Collaborative Filtering With Adaptive Neighbourhoods (Extended Version). In *Research Note RN/09/03, Dept. of Computer Science, University College London*, London, UK, April 2009.

[2]Available at http://www.cs.ucl.ac.uk/staff/n.lathia