Item Selection Strategies for Collaborative Filtering

Rachael Rafter, Barry Smyth
Smart Media Institutce, University College Dublin, Ireland
{rachae 1.rafter, barry.smyth } @ucd.ie

1 Introduction

Automated collaborative filtering (ACF) methods leverage the ratings-based profiles of users that are similar to some target user in order to proactively select relevant items, or predictively rate specific items, for the target user. Many of the advantages of ACF methods are derived from its content-free approach to recommendation; it is not necessary to rely on content-based descriptions of the recommendable items, only their ratings distribution across the population of raters. Furthermore, ACF methods have an element of serendipity associated with them, as users can find items for which they would never have explicitly searched but nonetheless find interesting. The raters, items and ratings form an sparse matrix, *R* and the classical ACF prediction task is to predict the rating that user *t* will give to item *i* given that R_{ltl} is currently empty. For example, Resnick's well-known algorithm [Resnick *et al,* 1994] predicts $R_{t,i}$ based on t's average rating, $\bar{t}$, and $R_{r,i}$, the rating each rater *r* gives to i, relative to r's average rating, $\bar{r}$, and weighted by $C_{t,r}$, the correlation between the shared ratings of t and r.

$$R_{t,i} = \bar{t} + \frac{\sum_{\forall r \neq t}((R_{r,i} - \bar{r}) * C_{t,r})}{\sum_{\forall r \neq t} |C_{t,r}|}$$

It is well known that the success of ACF depends broadly speaking, on the quality of the user profiles and on the coverage of the ratings space that they provide [Smyth *et al,* 2002; Rafter and Smyth, 2001a; Rafter *et al,* 2000a; 2000b; Herlocker *et al,* 1999; Breese *et al,* 1998; Konstan *et al,* 1997]. At the same time efficiency depends on the number of users that are taken into account during rating and the size of their individual profiles. For example, Resnick's algorithm is *0(r.i)* where r is the number of raters and *i* the average number of items that they have rated.

In this paper we are interested in looking for ways to improve the efficiency of ACF without compromising the quality of recommendations. Our basic proposal is to find ways of reducing the number of items that need to be considered during recommendation. For example, Resnick's algorithm above compares each rater profile to the target user by computing the correlation between all item ratings that these profiles share, which can often be a large number of ratings. By selecting a subset of shared ratings we can significantly improve the efficiency of this fundamental prediction step. However, naive item selection strategies are likely to result in significant reductions in recommendation accuracy and so more intelligent strategies, based on an understanding of the relative value of different items, are proposed.

2 Item Selection Strategies

Let us assume that for reasons of efficiency it is necessary to select a subset of *k* shared items for consideration during ACF prediction using Resnick's algorithm. Which subset should be chosen? One could select the *k* items at random. Alternatively one to look to identify those items that are likely to contribute more information to the prediction task.

2.1 Inverse Popularity

One approach to selecting useful items during ACF prediction is to look at how popular an item is among the rater population. The motivation here lies in the belief that relatively poorly distributed items among the population should be better predictors of a users preferences [Rafter and Smyth, 2001b]; if a user has looked at an unusual item this should tell us more about her preferences than an item that everyone is familiar with. For example in a movie recommendcr, most people will have seen movies like "The Lord of the Rings" or "Fight Clubhand therefore most profiles will contain these movies, but all this really tells us is that these are popular movies. However if a user profile contains more unusual movies such as "Run Lola Run" or "Buffalo 66", that most other profiles don't contain, then we can tell more about that user's *individual* taste for movies. Accordingly, our inverse popularity strategy weights the items in a profile according to the number of users in the population that have that item in their profile and selects those *k* items, shared between rater and target user, that have the highest inverse popularity score (Equation 2).

$$InvPop(i) = 1 - \frac{|r_j \epsilon R : r_j(i) \neq \{\}|}{|R|} \qquad (2)$$

2.2 Deviation of Ratings

The deviation of ratings strategy is based on the principle that the more users tend to disagree about an item the more informative that item is about any given user's particular preferences. For example, in a movie recommender most users will give a movie like "The Shawshank Redemption" a high rating and so there is little to be gained by comparing users along this dimension. However, by the same token, there may tend to be a lot of disagreement about "Vanilla Sky", and knowing whether a user likes or dislikes this movie is likely to tell us a lot about this user.

One way to take account of the above idea is to weight items according to the standard deviation of their ratings across the user population. Items that have a high standard deviation in their ratings (ri) (a lot of disagreement) get a high weight, and items with a lot of agreement (low standard deviation) get a low weight (Equation 3). Once again, using this strategy, we can prioritise the selection of the *k* shared items.

$$DevRating(i) = StdDev(i) = \sqrt{\frac{\sum_{\forall r_i \in R}(r_i - \bar{r})^2}{|R| - 1}} \quad (3)$$

3 Experimental Evaluation

An experimental evaluation has been carried out to evaluate how our inverse popularity and deviation of ratings selection strategies impact prediction quality. To do this we have implemented a version of Resnick's ACF algorithm and modified it to select different sized subsets of shared items as the basis for prediction. Four different item selection strategies are reported:

- *All* selects all shared items as per the standard Resnick approach.
- *Random* selects a random subset of A: shared items.
- *Popularity* selects a subset of *k* shared items according to the inverse popularity metric.
- *Deviation* selects a subset of *k* shared items according to the deviation of ratings metric.

In turn we evaluated the above methods using a subset of the largest 2050 profiles (2000 as the user population, and 50 as the target users) from the EachMovie dataset, [McJones, 1997]. From each target user profile we selected 10 items at random (500 in total) for which to predict scores measuring the quality of the prediction as the *average prediction error* by computing the difference between the predicted rating and the actual rating in the standard way. In addition, we vary the value of *k* from 5 to 100 to investigate the impact of different profile subset sizes.

The results are presented in Figure 1 as graphs of average prediction error versus *k* for each of the 4 selection strategies (All, Random, Popularity, Deviation). The results for All are obviously a straight line as this strategy in unaffected by *k* by design. The error for this strategy serves as an optimal benchmark against which to judge the other 3 techniques. In each case we find that the Popularity and Deviation techniques significantly outperform the naive Random selection approach. For example, at A: = 25 the Random method presents with a prediction error of about 0.207 compared to errors of 0.185 and 0.186 for the Popularity and Deviation methods, and a minimum error of 0.183 for the All benchmark. A similar pattern is found for k < 50.

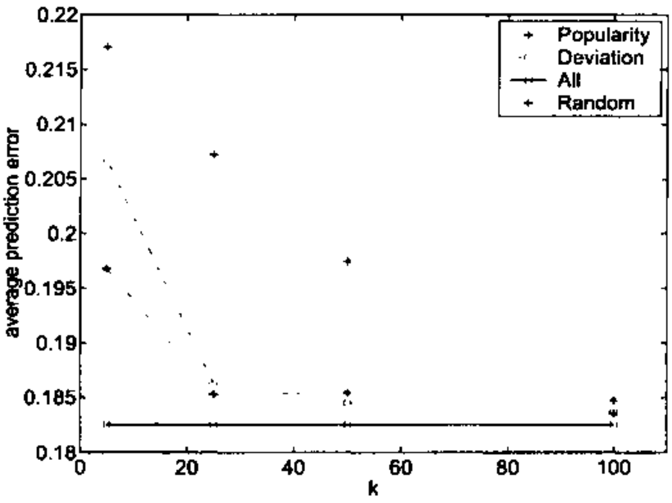

Figure 1: Average prediction error v. .v. *k*

It is interesting to note that for *k* = 100 the prediction accuracy of the various techniques is seen to converge, indicating that there is little difference in the selection benefits for item subsets of this size. However, the average overlap of the population is 116.75 meaning that at the *k* = 100 mark almost all overlapping items are being considered by any of the four strategies and therefore it is appropriate that the accuracy converges. In point of fact it is appropriate that item selection strategies work well at low values of *k* for two reasons. First, it maximises the potential efficiency advantages. Secondly, most profiles tend to have relatively small numbers of ratings and so the shared ratings between two profiles is likely to be small; in EachMovie more than 90% of profiles have less than 100 ratings.

4 Conclusions and Future Work

We have described two item selection strategies for ACF (Deviation and Popularity) that reduce the set of items used when making predictions. We have discussed how these strategies improve the efficiency of ACF and our evaluation has shown that these improvements incur little compromise in the quality of prediction, when compared with the standard Resnick algorithm. The results also show that our strategies outperform the naive Random strategy and therefore are selecting a more intelligent subset of items. Efficiency in ACF is a well documented problem and we believe that item selection strategies such as those proposed in this paper are one of the keys to solving efficiency problems. In the future we look to investigating other similar item selection strategies (such as combining the Popularity and Deviation methods), based on a further understanding of the relative values of items.

The development of item selection strategies such as those we have described forms part of a larger research goal concerning the value of items in ACF. In any ACF prediction or recommendation task we can imagine three distinct beneficiaries. The first and most obvious beneficiary is the target user, the person for whom the prediction is being made. The second is the party making the recommendation - for example, an online store whose desire it is to make recommendations to the target user and encourage them to purchase something. Finally the ACF system itself is also a potential beneficiary in the sense that with each prediction or recommendation made the system learns a new piece of information about the user. In such a situation where three distinct parties can potentially benefit from the outcome of the recommendation, there are clear trade-offs. For example, the online store may prefer to recommend expensive items over cheaper alternatives to customers and even weight items for recommendation not only according to their relevancy to the customer but also according to their price. Conversely, the user is unlikely to opt for such a recommendation strategy. The trade-off we are most interested in investigating is that of optimising recommendations for short-term benefit against optimising them for long-term benefit. The straightforward recommendation strategy is to select the item with the highest relevancy for the target user (i.e. the one she is most likely to like) for recommendation. This optimises recommendations on a short-term basis - the best option is selected for the current recommendation. However, if we were to weight recommendations not only based on the probability of the user liking them, but also based on the amount of knowledge the system is likely to gain with each recommendation we could potentially make better recommendations in the future.

Our research so far has shown that different items have different levels of knowledge gain. Relatively unpopular items, or those that tend to have a high deviation of ratings tell us more about the user and her preferences than an item that everyone has seen and liked. In the future we plan to look at the potential benefit of weighting items not only according to their relevance to the user, but also according to their associated knowledge gain, and investigating how such a strategy affects the quality of future predictions.

References

[Breese *et ai,* 1998] J. Breesc, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of 14th Conference on Uncertainty in Artificial Intelligence,* Madison, Wisconsin, USA, July 1998.

[Herlocker et al., 1999] J. Herlocker, J. Konstan, A. Borchers, and J. Reidl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Berkeley, California, USA, August 1999.

[Konstan *et al,* 1997] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Reidl. Grouplens: Applying collaborative filtering to Usenet news. *Communications of A*CM, 40(3):77-87, March 1997.

[McJones, 1997] P. McJones. Eachmovie collaborative filtering data set. Available from http://research.compaq.com/SRC/eachmovie/, 1997.

[Rafter and Smyth, 2001a] R. Rafter and B. Smyth. Passive profiling from server logs in an online recruitment environment. In *Proceedings ofIJCAI Workshop on Intelligent Techniques for Web Personalization (ITWP2001),* Seattle, Washington, U.S.A., August 2001.

[Rafter and Smyth, 2001b] R. Rafter and B. Smyth. Towards a domain analysis methodology for collaborative filtering. In *Proceedings of 23rd European Colloquium on Information Retrieval Research (ECIROI),* Darmstadt, Germany, April 2001.

[Rafters *al,* 2000a] R. Rafter, K. Bradley, and B. Smyth. Automated collaborative filtering applications for online recruitment services. In *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems,* Trento, Italy, August 2000.

[Rafter et al., 2000b] R. Rafter, K. Bradley, and B. Smyth. Personalised retrieval for online recruitment services. In *Proceedings of the 22nd Annual Colloquium on Information Retrieval (IRSG2000),* Cambridge, UK, April 2000.

[Resnick *et al,* 1994] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Reidl. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM Conference on Computer-Supported Cooperative Work (CSCW94),* Chapel Hill, North Carolina, USA, August 1994.

[Smyth*etal,* 2002] B. Smyth, K. Bradley, and R. Rafter. Personalization techniques for online recruitment services. *Communications of the ACM,* 45(5):39^0,2002.