Re-Considering Neighborhood-Based Collaborative Filtering Parameters in the Context of New Data

Adele E. Howe
Computer Science Department
Colorado State University
Fort Collins, CO 80523, U.S.A.
howe@colostate.edu

Ryan D. Forbes
ReadyTalk
1598 Wynkoop St.
Denver, CO 80202, U.S.A.
ryan.forbes@readytalk.com

ABSTRACT

The Movielens dataset and the Herlocker et al. study of 1999 have been very influential in collaborative filtering. Yet, the age of both invites re-examining their applicability. We use Netflix challenge data to re-visit the prior results. In particular, we re-evaluate the parameters of Herlocker et al.'s method on two critical factors: measuring similarity between users and normalizing the ratings of the users. We find that normalization plays a significant role and that Pearson Correlation is not necessarily the best similarity metric.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval, Information Filtering

General Terms

Experimentation

1. INTRODUCTION

The algorithmic basis for some of the most commonly used and influential collaborative filtering (CF) methods was developed and first evaluated around 1999 using the Movielens data set [3]. From this study, the preferred method[1] was neighborhood-based using Pearson correlation coefficient for the similarity measure, choosing the k most similar users for neighborhood generation, and the "deviation from mean" method for normalization. A later study [2] recommended deviation from mean or z-score for normalization.

In October 2006, a new collaborative filtering data set was introduced to the CF community: the Netflix Challenge set. This new data set provides an opportunity to re-visit the conclusions about neighborhood-based collaborative filtering drawn in earlier studies using the Movielens data. We conducted a factorial study varying the parameters of the HKBR method to assess their impact on performance for Movielens as well as subsets of the Netflix challenge set. In particular, we study parameter choices for similarity metrics and normalization schema and examine differences between the Movielens and Netflix data sets to establish the generality of results, gauge the appropriateness of the HKBR method as a baseline in comparative studies, and examine the effects of the data sets.

[1]We will refer to this as 'HKBR method' after the authors.

CIKM'08, October 26–30, 2008, Napa Valley, California, USA.
ACM 978-1-59593-991-3/08/10.

2. EXPERIMENT DESIGN

Given similarity measure s, neighborhood selection method h, normalization scheme n, and ratings R, the rating user a gives item i is predicted with the following algorithm.

```
Predict(a, i, s, n, h, R)
1. R = n(R)          # Normalize R according to n
2. Generate users, U s.t. R_{u,i} ≠ 0 and u ≠ a, ∀u ∈ U
3. For each u ∈ U
     Sim_u = s(u,a) # Compute similarity of u to a
4. H = h(U, Sim)    # Construct the neighborhood
5. return n^{-1}(Σ_{u∈H} R_{u,i}/|H|)  # De-normalize
```

Steps 2 - 4 are the same as [3]. Steps 1 and 5 differ in how the normalization is applied. These steps allow for normalization before computing similarity (*pre-normalization*); HKBR normalized during prediction (*post-normalization*). The variable settings tested were {25, 50} for h, {pre and post normalization} for normalization timing, {none, mean-centering (`mc`), unit standard deviation (`usd`) and zero mean / unit standard deviation (`zmus`)} for n and {cosine vector similarity, Pearson correlation, Spearman correlation, Manhattan, Euclidean, L_∞ and Hamming} for s.

For our analyses, we set normalization timing and h parameters from the empirical results. In our results, normalization timing exerts no effect, except that pre-normalization improves CVS. So, we use pre-normalization. For h, we find that 25 works best for Movielens and 50 for Netflix.

We used two datasets: Netflix (`http://www.netflixprize.com/`) and Movielens (`http://www.cs.umn.edu/research/grouplens`). We produced 10 subsets of Netflix following the procedure of the earlier study while trying to match the Movielens set for two of the statistics: number of items (fixed at 1776 ± 1) and number of ratings (912552 ± 130774). We constructed an additional Netflix subset (Dense) to examine what happens when a key statistic (mean ratings per user) is very high (1279 versus 106 for Movielens and 30 for the subsets). The datasets were divided into training and testing using a method similar to that of [3]. Ten percent of users were randomly selected for the test set; for each such user, five ratings were withheld.

3. RESULTS AND ANALYSIS

We address three primary questions about the conclusions of the prior studies in our experiments.

3.1 Are there interaction effects?

The results clearly show interaction effects, especially with the Netflix data subsets. For Movielens, an Analysis of Variance (ANOVA) test to check for interaction between similarity and normalization shows significant main and interaction effects when the input includes `none` and `usd` normalization, but only a main effect of similarity when `none` and `usd` are removed. Pearson and Spearman correlation only perform well on either data set when the data are mean-centered (with `mc` or `zmus`). The results also show that when the users are normalized with `mc`, CVS performs significantly better than Pearson correlation (t-test on Movielens yields $p \leq 1.23 \times 10^{-07}$, on Netflix $p \leq 0.02$).

3.2 Do our 'best' parameter choices differ from previous?

For normalization, Herlocker et al. found no significant difference between `mc` and `zmus` [3], but t-tests on our results show that using the pre-normalize method and `zmus` provides statistically significantly ($p < 10^{-10}$) better performance than `mc` or the other normalization schemes.

For similarity, Breese et al. found that Pearson correlation performed better than cosine vector similarity [1]. Our results show that on the Movielens data set, measuring similarity with CVS performs significantly better than using Pearson correlation, no matter which normalizer is used. The improvement is very small, but is highly significant.

Surprisingly, we found that Hamming similarity with no normalization outperforms all of the other methods on the Netflix subsets. Yet, this is the combination that performed the worst on the Movielens data. Also, Hamming similarity is the only similarity measure that performed better on either data set with no normalization than with the mean-centering scheme.

3.3 Do results on Movielens generalize to Netflix?

We find some differences in performance of parameters when tested with Movielens and Netflix data subsets. In particular, for Netflix,

- generally the MAEs were higher,
- the preferred number of neighbors was higher,
- the interaction effects between similarity and normalization were stronger, and
- Hamming similarity with no normalization was the best performing (but worst for Movielens).

For both, generally, normalization improves the performance with `zmus` as the best scheme.

We used the Dense subset to follow-up on the difference in preferred parameterizations. As one would expect, the MAEs are, with a few exceptions, significantly lower for the Dense subset. The best performing parameters are Cosine similarity with `zmus` or `mc` normalization; Pearson and Spearman similarity with some normalization perform just slightly worse. Hamming similarity performs better on the dense subset than on the random subsets.

4. OBSERVATIONS

The Movielens dataset and the HKBR method have been very influential in comparative evaluation of CF methods as benchmark dataset and baseline method, respectively. We assessed some of the recommendations from earlier studies in light of new alternative parameterizations and a new substantial dataset for the same application. We found that many of the prior recommendations do not generalize to the new dataset and that the parameters exhibit significant, but previously inadequately explored interaction effects. In particular,

- Normalization plays a significant role.
  - Pearson and Spearman similarity metrics only perform well with mean centering (`mc` or `zmus`).
  - Pre-normalization with `zmus` produces better performance than other normalization schemes.
- Different data sets (Movielens, Netflix subsets and Netflix Dense) favor different parameterizations.
- The HKBR method may not be the best baseline method for comparison, and CVS should not have been so readily abandoned.

We note though that many of the observed differences, although statistically significant, are none the less small. Improving the mean absolute error by 0.01 on the data sets we used means that the average user will see their movie rating predictions improve by 0.01 stars. Such a small improvement may not be noticed by the average user.

Although our observations complicate comparative evaluation, they also afford opportunities to better understand the evolution of Collaborative Filtering and of what makes the methods successful.

5. REFERENCES

[1] J. S. Breese, D. Heckerman, and C. M. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI '98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52, Madison, Wisconsin, USA, 1998.

[2] J. Herlocker, J. A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5:287–310, 2002.

[3] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237, New York, NY, USA, 1999. ACM Press.

[4] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.

[5] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508, New York, NY, USA, 2006. ACM Press.