Imputed Neighborhood Based Collaborative Filtering

Xiaoyuan Su
Computer Science & Engineering
Florida Atlantic University
xsu@fau.edu

Taghi M. Khoshgoftaar
Computer Science & Engineering
Florida Atlantic University
taghi@cse.fau.edu

Russell Greiner
Department of Computing Science
University of Alberta
greiner@cs.ualberta.ca

Abstract

Collaborative filtering (CF) is one of the most effective types of recommender systems. As data sparsity remains a significant challenge for CF, we consider basing predictions on imputed data, and find this often improves performance on very sparse rating data. In this paper, we propose two imputed neighborhood based collaborative filtering (INCF) algorithms: imputed nearest neighborhood CF (INN-CF) and imputed densest neighborhood CF (IDN-CF), each of which first imputes the user rating data using an imputation technique, before using a traditional Pearson correlation-based CF algorithm on the resulting imputed data of the most similar neighbors or the densest neighbors to make CF predictions for a specific user. We compared an extension of Bayesian multiple imputation (eBMI) and the mean imputation (MEI) in these INCF algorithms, with the commonly-used neighborhood based CF, Pearson correlation-based CF, as well as a densest neighborhood based CF. Our empirical results show that IDN-CF using eBMI significantly outperforms its rivals and takes less time to make its best predictions.

1. Introduction

Many successful recommender systems use collaborative filtering *(CF),* which uses a database of user preferences of items to predict additional topics or products that other users might like. A typical *CF* scenario involves a "ratings matrix" over m users $\{u_1, u_2, \ldots, u_m\}$ and n items $\{i_1, i_2, \ldots, i_n\}$, whose (u,i) entry denotes how user u rated item i. The ratings can either be explicit indications (e.g., on a 1-5 scale) or implicit indications, such as purchases or click-throughs [6]. Note this matrix is typically very sparse, as most users rate relatively few items.

Existing *CF* solutions roughly fall in the following three categories. Memory-based *CF* algorithms use the entire rating data (or perhaps a subsample) to generate a prediction. Model-based *CF* algorithms first build models, such as machine learned classifiers, from user ratings, then make predictions using the models. Hybrid *CF* algorithms make predictions or recommendations typically by combining *CF* and content-based recommenders, which are based on auxiliary information, such as documents, URLs, news messages, web logs, user profiles and item descriptions about users' tastes, preferences, and needs.

The Pearson correlation-based *CF* (*Pearson CF*) algorithm is a representative memory-based *CF* algorithm [8], which calculates similarities between each pair of items rated by a user, or between each pair of users who rate the same item (Section 2). This allows these systems to scale effectively with the number of users and items, as this computation involves only a small percentage of the total number of items and users – i.e., due to the sparsity. One shortcoming of *Pearson CF* is that its predictive performance degrades quickly as the data becomes more sparse [9]. Neighborhood-based *CF* makes *CF* predictions using *Pearson CF* based on the nearest neighbors instead of the whole dataset; Bell and Koren [1] demonstrated that this improved predictive performance.

Model-based *CF* techniques, such as Bayesian belief nets *CF* algorithms, address the sparsity challenge of *CF* tasks by providing relatively accurate predictions even given sparse rating data. However, their empirical performance is not significantly better than the *Pearson CF* or is not good enough [7][9].

In order to improve predictive accuracy for *CF* tasks by taking advantage of additional content information, *content-boosted CF* (a representative hybrid *CF* method) uses *naïve Bayes* as the predictor to estimate missing values based on content information and applies a *weighted Pearson CF* algorithm to produce *CF* predictions on the generated *pseudo rating matrix* (the rating matrix whose missing values have been filled in) [5].

This *Content-boosted CF* system begins by filling in the missing values; this corresponds to "imputation",

which is commonly used in application areas, such as social surveys, where missing data may cause difficulties in estimation and inference [11]. Figure 1(b) presents an example of imputed data – a *pseudo rating matrix* – which can be used as the input for *CF* prediction.

	U_1	U_2	U_3	U_4	U_5
I_1			4		1
I_2	2		2	2	2
I_3	?	1		2	4
I_4	3	3	3	2	3
I_5	1		1		

U_1	U_2	U_3	U_4	U_5
2	3	4	3	1
2	2	2	2	2
2	1	3	2	4
3	3	3	2	3
1	1	1	1	2

Figure 1. (a) original rating data, (b) imputed rating data (pseudo-rating matrix)

As imputation techniques can generally improve performance when mining incomplete data, we observe that *CF* predictions based on imputed data will also improve predictive performance for *CF* tasks.

We previously proposed *imputation-boosted collaborative filtering* (*IBCF*) systems that impute the missing values with machine learned classifiers or mean imputation before using the *Pearson CF* to make final predictions on the imputed data. Our earlier results show that these systems improved predictive performance on low dimensional rating data (that have a small number of users or items) [12][13]. However, we did not apply the *IBCF* approach to the commonly used neighborhood based *CFs* [8][1].

We therefore propose two imputed neighborhood based collaborative filtering (*INCF*) algorithms: imputed nearest neighborhood *CF* (*INN-CF*), which first finds the users most similar to the active user (for which we are making predictions), then uses the corresponding imputed rating data to make predictions for the active user; and imputed densest neighborhood *CF* (*IDN-CF*), which makes predictions from the imputed densest neighbors (i.e., the users who have rated the most number of items). The imputation techniques we considered include the baseline mean imputation (*MEI*) and an extension of Bayesian multiple imputation (*eBMI*). To evaluate our systems, we compare these *INCF* algorithms with the commonly-used *Pearson CF*, neighborhood-based *CF* algorithm, and a densest neighborhood based *CF* algorithm.

As performance metrics, we use *mean absolute error* (*MAE*), which computes the average of the absolute difference between the predictions and true ratings:

$$MAE = \frac{\sum_{\{u,i\}} \left| p_{u,i} - r_{u,i} \right|}{n} \quad (1)$$

where n is the total number of ratings over all [*user, item*] pairs in the test set, $p_{u,i}$ is the predicted rating for user u on item i, and $r_{u,i}$ is the ground truth rating value. The lower the *MAE*, the better the predictor.

Section 2 presents Pearson correlation based *CF* and neighborhood based *CF* algorithms. Section 3 presents the framework of our imputed neighborhood based collaborative filtering algorithms. Section 4 presents experimental design and results.

2. Pearson correlation based CF and neighborhood based CF algorithms

2.1 Pearson correlation based CF algorithm

The traditional Pearson correlation based collaborative filtering algorithm, *Pearson CF*, is a representative memory-based *CF* algorithm that uses the rating database to generate a prediction.

The *Pearson correlation* between item i and j is

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}} \quad (2)$$

where the $u \in U$ summations are over the users who rated both the items i and j , and $\bar{r}_i$ is the average rating of the co-rating users on the i-th item. In Figure 1(a), for $i=I_3$ and $j=I_4$, we find $\bar{r}_3 = 2.33$ by computing the average rating on the item I_3 from the three users that also rated item I_4 (who are U_2, U_4, and U_5), and $\bar{r}_4 = 2.67$, and the *Pearson correlation* between I_3 and I_4 is $w_{3,4} = 0.189$. We similarly compute $w_{i,j}$ for all pairs of items i and j.

To make a prediction for the active user, a, on a certain item, b, the item-based *Pearson CF* algorithm returns the weighted average of all the ratings on that item using the formula

$$p_{a,b} = \bar{r}_b + \frac{\sum_{i \in I} (r_{a,i} - \bar{r}_i) \cdot w_{b,i}}{\sum_{i \in I} \left| w_{b,i} \right|} \quad (3)$$

where $\bar{r}_b$ and $\bar{r}_i$ are the average ratings for item b and item i that all other users have rated, and $w_{b,i}$ is the *Pearson correlation* between items b and i, defined in Equation 2 above [2]. The $i \in I$ summations are over all the items that the user a has rated. Here in Figure 1(a), for user $a=U_1$, and item $i=I_3$, the prediction is $p_{1,3} = 3$.

2.2 Neighborhood based CF algorithm using adjusted Pearson correlation

The "neighborhood based *CF*", aka the *nearest neighborhood based CF* algorithm, first finds the N neighbors nearest to the active user, then makes *CF* predictions using *Pearson CF* based only on those neighbors.

To find the nearest (most similar) neighbors of the active user u, we first calculate the similarity in terms of *Pearson correlation* between user u and each other user v

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}} \quad (4)$$

where the $i \in I$ summations are over the items that rated by both the users u and v and $\bar{r}_v$ is the average rating of the co-rated items of the v-th user. To help regularize the values, and also avoid the divide-by-zero (*DBZ*) situation (when all of the values are the same, and so the variance is 0), we use an *adjusted Pearson correlation*:

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} + \varepsilon - \bar{r}_u)(r_{v,i} + \varepsilon - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} + \varepsilon - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} + \varepsilon - \bar{r}_v)^2}} \quad (5)$$

where ε is a small value. For example, in Figure 1(a), with ε=0.001, the *adjusted Pearson correlation* between u_4 and u_5 on co-rated items (between {2,2,2} and {2,4,3}) will be 0.002 instead of a *DBZ* exception. This *adjusted Pearson correlation* will slightly bias the original Pearson correlation for normal cases. For example, the *adjusted Pearson correlation* (from Equation 5) for {3,3,4} and {5,4,4} is -0.499996, while the original Pearson correlation (Equation 4) is -0.5. We similarly compute *adjusted Pearson correlation* $w_{u,v}$ for all pairs of users u and v, and take the N users with the top values of $w_{u,v}$ as the most similar neighbors of the active user u.

Note we use the *adjusted Pearson correlation only* for the divided-by-zero cases and otherwise use the original *Pearson correlation*. Our empirical results show that this strategy performs better than solely using the original *Pearson correlation*, or solely using the *adjusted Pearson correlation*.

As the *neighborhood based CF* is effective in producing *CF* predictions, it is commonly used and deployed in many commercial systems. A major shortcoming of *neighborhood based CF,* however, is its poor predictive performance when the rating data is sparse. This is especially true for new users, who have very few ratings, as here the set "I" in Equation (4) will be small, or even empty. We address this by using imputation: rather than run this *neighborhood based CF* algorithm on the original data, our *INCF* algorithms (described in the following sections) first impute values for the missing entries based on the most similar or the densest users, then run *Pearson CF* on the imputed data to make *CF* predictions. As *Pearson CF* typically has better predictive accuracy for denser rating data, we expect this will lead to better performance. However, this means the overall algorithm will be slower than neighborhood based *CF* over the same number of neighbors, as its predictions are based on the (pseudo) rating matrix (Figure 1(b)), which is now much denser. Nevertheless, *INCF* algorithms may require less time to produce predictions as they might need much fewer neighbors.

2.3 Densest neighborhood based CF algorithm

Our novel densest neighborhood based *CF* (*DNCF*) makes *CF* predictions based on the densest users (those who have rated most number of items), rather than the most similar users. *DNCF* is inspired by the fact that *Pearson CF* performs better on denser rating data.

This algorithm first ranks the users based on the number of items they have rated, then uses *Pearson CF* on the ratings of the top N densest users to give *CF* predictions for the active user. Note that these densest users does not depend on the active user

3. Imputed neighborhood based CF

The main steps of imputed neighborhood based *CF* algorithm are (see Figure 2): (1) impute the user rating data to generate a pseudo rating data (imputed data); (2) for each active user, find the N most similar users in the original rating matrix (for imputed nearest neighborhood *CF*, or *INN-CF*), or find the N densest users of the dataset (for imputed densest neighborhood *CF*, or *IDN-CF*); (3) make *CF* predictions for the active user by applying the traditional *Pearson CF* algorithm on the imputed rating data of the most similar (or the densest) users .

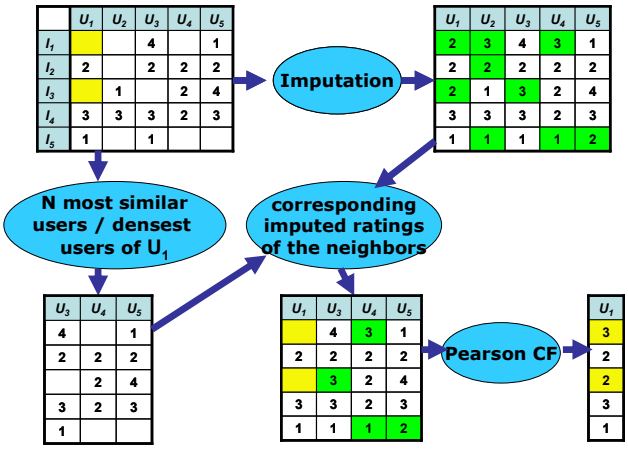

Figure 2. Illustration of imputed neighborhood based collaborative filtering: INN-CF (using most similar neighbors) and IDN-CF (using densest ones)

3.1 Imputation technique selection

The selection of an imputation technique should be based on an analysis of the missing data patterns. If the missingness does not depend on the observed data then this is considered *MCAR* (Missing Completely at Random, e.g., $r_{u,i}$ is missing with an iid (independent and identical distribution) probability of 0.6, independent of everything else). If the missing data depend on the observed data, but not on the unobserved data, they are *MAR* (Missing at Random, e.g., missingness on item I_m is dependent on observed ratings on item I_n, such as the entry *(u,m)* for a user's rating of a horror movie is missing if the value $r_{u,n}$ for that user's rating for *Bambi* is specified and is 5) [11]. Both *MCAR* and *MAR* missing data patterns are considered *ignorable*, as one can make valid likelihood-based inferences and estimations even if the missing data mechanism is ignored (not explicitly considered in the analysis). If the "missingness" is not *MAR*, then we say the data are *NMAR* (not missing at random) aka *non-ignorable*, in which patterns of the unobserved variables can determine their own missingness (e.g., $r_{u,i}$ is missing when it is <3). When the missing data are *NMAR*, a model of missing value mechanism must be incorporated into the estimation process, which is practically difficult, because the missing data mechanism is rarely known with certainty [10].

While most assume the missing pattern of *CF* data is *MAR*, there are also cases that they are *NMAR* -- eg, many users will not rate a movie because he did not like it. Here, a multiple imputation technique (see Section 3.3) can be used for *CF* tasks, as it typically reduces bias for both *MAR* and *NMAR* data. These motivate us to investigate applying a multiple imputation technique in the *INCF* framework.

Our previous work also investigated applying other imputation techniques for *CF* tasks, such as using linear regression imputation, and predictive mean matching [12]. However, as their performances are not as good as the techniques shown here, we do not consider them in this work.

3.2 INCF-MEI: imputed neighborhood based CF using mean imputation

INCF-MEI (*INCF using mean imputation*) produces the pseudo rating data using mean imputation on *users*, which replaces each missing value *(u,i)* in the rating matrix with the mean of the observed ratings on each user, that is $\frac{1}{k}\sum_{i \in I(u)} r_{u,i}$, where k is the total number of items $i \in I(u)$ that have been rated by the user u. Mean imputation is one of the simplest imputation technique. Unfortunately, it can distort the shape of distributions (by creating a spiked distribution at the mean in frequency distributions), and it also reduces the variance of the predictions.

3.3 INCF-eBMI: imputed neighborhood based CF using extended Bayesian multiple imputation

Multiple imputation (*MI*) produces many different imputed datasets – e.g., one such dataset might correspond to Figure 1(b), while another would fill in the missing values differently, perhaps setting r(U_2, I_2) to 5 (not 2), and r(U_3, I_3) to 2. Each set of new values is estimated independently from the posterior predictive distribution. The final estimate is obtained by combining these m estimates. Multiple imputation is better than standard single imputation as it accounts for the uncertainty about the predictions of the imputed values.

Bayesian multiple imputations (*BMI*) are created through a Bayesian process: first specify a parametric model for the complete data and a model for the mechanism by which data become missing (e.g., *MAR*); typically using some prior distribution over the unknown model parameters, then simulated m independent draws from the conditional distribution of the missing data given the observed data. In non-trivial applications, special computational techniques such as Markov chain Monte Carlo (*MCMC*) must be applied to create *BMI*. (While this *BMI* model assumes the imputations are generated from a multivariate normal distribution, *BMI* is known to be robust to departures from the assumption [10]).

In more detail, the *BMI* imputed data are created through the following steps [10]:

We assume that $Y = (Y_{obs}, Y_{miss})$ follows a parametric model $P(Y|\theta)$ where θ has the prior distribution $P(\theta)$; hence the posterior predictive distribution for Y_{miss} is

$$P(Y_{miss} \mid Y_{obs}) = \int P(Y_{miss} \mid Y_{obs}, \theta) P(\theta \mid Y_{obs}) d\theta \qquad (6)$$

BMI repeats the two-step process for $j=1, \ldots, m$:

(1a) draw $Y_{miss}^{(j+1)}$ from $P(Y_{miss} | Y_{obj}, \theta^{(j)})$;

(1b) draw $\theta^{(j+1)}$ from $P(\theta \mid Y_{obs}, Y_{miss}^{(j+1)})$.

(2) augment data using Markov Chain Monte Carlo (*MCMC*), in which the following Markov chain $\{Y_{miss}^{(1)}, \theta^{(1)}, Y_{miss}^{(2)}, \theta^{(2)}, \ldots, Y_{miss}^{(j)}, \theta^{(j)}, \ldots\}$ is generated, where each pair $\langle Y_{miss}^{(j)}, \theta^{(j)} \rangle$ depends on the previous one, and the stabilized distribution ($\langle Y_{miss}^{(j)}, \theta^{(j)} \rangle$ as $j \to \infty$) is a draw from the target probability distribution function.

These two steps are repeated enough times between successive entries that (we anticipate) the values have no additional correlations [11].

After producing m sets of imputed data, we take the average as the final imputed data

$$\bar{\theta} = \frac{1}{m}\sum_{i=1}^{m}\hat{\theta}_i \quad (7)$$

where $\hat{\theta}_i$ is the estimate based on the *i*-th imputed data.

In the cases when the entries for an instance is empty (ie., there is no ratings on a new item, from *any* user), *BMI* will be unable to produce any imputed and so leave those entries missing again. We therefore propose an extension of Bayesian multiple imputation (*eBMI*) to handle this problem, by filling in these missing values using the attribute average (mean imputation). Hence,

$$\bar{\theta}_{eBMI} = \begin{cases} \bar{\theta}_{BMI} & (\bar{\theta}_{BMI} \neq \phi) \\ \bar{\theta}_{MEI} & (\bar{\theta}_{BMI} = \phi) \end{cases} \quad (8)$$

where $\bar{\theta}_{eBMI}$, $\bar{\theta}_{BMI}$, $\bar{\theta}_{MEI}$ are imputed values from *eBMI*, *BMI*, and *MEI* respectively: that is, when the imputed value from *BMI* is empty (ϕ), *eBMI* uses $\bar{\theta}_{MEI}$ as the imputation; otherwise, it uses $\bar{\theta}_{BMI}$.

As the *eBMI* values must be within the attribute's range [*min*, *max*], we replace an estimate < *min* with *min*, and any >*max* with *max*.

$$\bar{\theta} = \begin{cases} \min & (\bar{\theta} < \min) \\ \bar{\theta} & (\bar{\theta} \in [\min, \max]) \\ \max & (\bar{\theta} > \max) \end{cases} \quad (9)$$

4. Experimental design

We worked on the *MovieLens* dataset [4], which is a widely used movie recommendation dataset with 100,000 ratings for 1682 movies by 943 users, where each rating is in {1,2,..,5}, where 93.7% of the entries are missing. Numerous research papers on collaborative filtering are based on this dataset [3][8]. We use 20% of the 100,000 ratings of *MovieLens* dataset as test data, and use the remaining 80% observed ratings as the training data; this training-test split is used by many researchers working on this dataset [8].

We implement the proposed imputed neighborhood based *CF* algorithms, *INN-CF* and *IDN-CF* (using *eBMI* and *MEI* as the imputers – corresponding to 4 different algorithms), the densest neighborhood based *CF* algorithm (which uses only the original data with no imputation), as well as the standard neighborhood based *CF* algorithm, and *Pearson CF*. We experimented on the above *CF* algorithms with neighborhood size ranging from 20 through 940, with a step-size of 20 neighbors (except for *Pearson CF, which* uses the possible maximum neighbors, which is *total user number - 1*), and investigate their respective predictive performance in terms of *MAE* (Equation 1).

The experimental results on the original *MovieLens* data are shown in Table 1, Table 2, and Figure 3 (testing on 19,996 reserved ratings out of 100,000).

Table 1. MAE performance of the CF algorithms on the MovieLens dataset

CF algorithms	best MAE	neighbor # at best	time (m)
densest neighborhood CF	0.7151	920	5
neighborhood-based CF	0.7097	620	5
INN-CF (eBMI)	0.7133	460	6
INN-CF (MEI)	0.7736	520	7
IDN-CF (eBMI)	**0.6926**	120	2
IDN-CF (MEI)	0.7440	60	1
Pearson CF	0.7205	942	7

Table 2. Part of the MAE performances of the CF algorithms with different neighborhood sizes on the MovieLens dataset

Neibr #	DNCF	NBCF	inn-cf (eBMI)	idn-cf (eBMI)	idn-cf (MEI)
60	0.7469	0.9396	0.7856	0.6996	**0.7440**
120	0.7334	0.8510	0.7605	**0.6926**	0.7519
180	0.7276	0.7766	0.7411	0.6950	0.7544
240	0.7257	0.7426	0.7282	0.6984	0.7596
300	0.7233	0.7293	0.7202	0.6989	0.7620
360	0.7204	0.7228	0.7157	0.6992	0.7644
420	0.7207	0.7184	**0.7143**	0.7023	0.7674
480	0.7192	0.7125	0.7145	0.7048	0.7696
540	0.7176	0.7119	0.7141	0.7089	0.7720
600	0.7170	**0.7098**	0.7158	0.7094	0.7729
660	0.7165	0.7126	0.7179	0.7118	0.7742
720	0.7159	0.7147	0.7191	0.7155	0.7753
780	0.7164	0.7140	0.7210	0.7179	0.7765
840	0.7164	0.7139	0.7220	0.7206	0.7780
900	**0.7159**	0.7152	0.7248	0.7235	0.7794
940	0.7159	0.7152	0.7258	0.7255	0.7805
Ave	0.7238	0.7504	0.7292	0.7071	0.7665

Imputed densest neighborhood *CF* (*IDN-CF*) using *eBMI* has the best performance here, with a 3.9% lower *MAE* than the *Pearson CF* (0.6926 vs. 0.7205). Moreover, it needs fewer neighbors (with 120 neighbors, out of a possible maximum size of 942) to reach the best performance, than most of its rivals (which need more than 450 neighbors each, except *IDN-CF* using *MEI,* which needs only 60 neighbors to achieve the best *MAE* but has much worse

performance). With a smaller neighborhood, *IDN-CF* using *eBMI* produces *CF* predictions in less time (2 minutes), in comparison with other *CF* algorithms (except *IDN-CF* using *MEI*) on a computer with Xeon E5430QC/2.66G CPU and 32.8GB RAM. Imputed nearest neighborhood *CF* (*INN-CF*) using *eBMI* performs better than the *Pearson CF* with a 1% lower *MAE*. Both *IDN-CF* and *INN-CF* using *eBMI* significantly outperform the commonly-used neighborhood based *CF*, with 5.8% and 2.8% lower average *MAE* scores over all the neighborhood sizes we investigated, and with 1-sided t-test p-values of $p<0.0002$ and $p<0.003$ respectively. Our densest neighborhood based *CF* also outperforms the (nearest) neighborhood based *CF* with a 3.5% lower average *MAE* score, with $p<2E-5$.

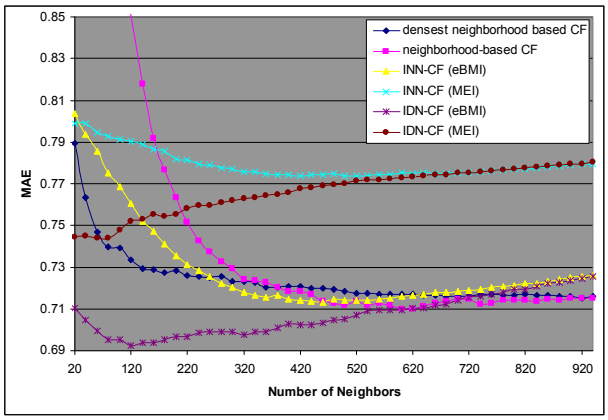

Figure 3. MAE performance of the INCF algorithms on the MovieLens dataset

Our *INCF* algorithms (*IDN-CF* using *eBMI* and *INN-CF* using *eBMI,* which have the best *MAE* scores of 0.6926 and 0.7133 respectively) beat the best *MAE* score on the same *MovieLens* dataset in literature [3], which applied an item-based *Pearson CF* with an *MAE* of 0.72 [8].

Both *IDN-CF* and *INN-CF* perform gradually worse with more neighbors than the neighbors producing the best performance, probably due to overfitting.

Mean imputation (*MEI*), the baseline imputation technique, did not improve *CF* performance when applied to either *IDN-CF* or *INN-CF*. As *MEI* will replace each missing value with that attribute's mean value, if the *CF* rating data is missing many values for an attribute, then essentially all of those missing values will become that mean value. It is especially problematic for datasets with *few items*, such as *MovieLens*, where 40% of the users rated five or fewer items; moreover, by holding out 20% of observed ratings, the users had yet fewer ratings.

Although *IDN-CF* using *eBMI* improved predictive performance and reduced online prediction time, it needed more time for offline imputation. When imputing the training dataset using *eBMI*, we worked on segmented smaller datasets, each of which had 100 users (attributes) except the last one that has 43; and later aggregated the imputed datasets into one. This strategy reduces the offline imputation time, although each subset still required about 11 minutes to impute using our computer.

Other advanced imputation techniques may also be effective in our imputed neighborhood based *CF* algorithms. We plan to investigate this in the future.

5. Conclusions

Collaborative filtering (*CF*) is one of the most successful approaches to building effective recommender systems. However, these *CF* algorithms must be able to deal with high-dimensional and very sparse rating data. Imputed neighborhood based *CF* (*INCF*) algorithms attempt to boost the performance of the standard *Pearson correlation-based CF (Pearson CF)* and *neighborhood based CF* systems by first computing, and then using, high-quality imputed data computed from the densest neighbors or the users most similar to the active user. We proposed a high quality imputation technique, an extension of Bayesian multiple imputation (*eBMI*), and found it is effective in the *INCF* framework, especially for the imputed densest neighborhood *CF* (*IDN-CF*) algorithm. *IDN-CF* using *eBMI* significantly outperforms the traditional *Pearson CF*, neighborhood based *CF* algorithms, and other *INCF* algorithms, and takes less online prediction time than most of its rivals. Imputed nearest neighborhood *CF* (*INN-CF*) using *eBMI* also has improved predictive performance for *CF* tasks.

References

[1] Bell, R., and Koren, Y. Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights, *IEEE International Conference on Data Mining (ICDM'07)*, 2007.

[2] Breese, J., Heckerman, D., and Kadie, C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *Proceedings of Fourteenth Conference on Uncertainty in Artificial Intelligence,* 1998.

[3] Candillier, L., Meyer, F., and Boulle, M. Comparing State-of-the-Art Collaborative Filtering Systems, P. Perner (Ed.): *MLDM*, *LNAI 4571*, pp. 548–562, 2007.

[4] GroupLens. 2006. http://movielens.umn.edu, *GroupLens Research group*, Department of Computer Science and Engineering, University of Minnesota.

[5] Melville, P., Mooney, R.J. and Nagarajan, R. Content-Boosted Collaborative Filtering for Improved

Recommendations, *Proceedings of the 18th National Conference of Artificial Intelligence*, 2002.

[6] Miller, B.N., Konstan, J.A. and Riedl, J. PocketLens: Toward a Personal Recommender System, *ACM Transaction on Information Systems*, 22(3), pp. 437-476, 2004.

[7] Miyahara, K., and Pazzani, M.J. Improvement of Collaborative Filtering with the Simple Bayesian Classifier, *Information Processing Society of Japan*, 43(11), 2002.

[8] Sarwar, B.M., Karypis, G., Konstan, J.A. and Riedl, J. Item-based Collaborative Filtering Recommendation Algorithms, *10th International World Wide Web Conference*, ACM Press, pp. 285-295, 2001.

[9] Su, X. and Khoshgoftaar, T.M. Collaborative Filtering for Multi-class Data Using Belief Net Algorithms, *the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pp 497-504, 2006.

[10] Rubin, D.B. Multiple Imputation for Nonresponse in Surveys. *J. Wiley & Sons*, New York, 1987.

[11] Schafer, J.L. Analysis of Incomplete Multivariate Data, *New York: Chapman and Hall,* 1997.

[12] Su, X., Khoshgoftaar, T.M., and Greiner, R. A Mixture Imputation-Boosted Collaborative Filter, *the 21st Conference of Florida Artificial Intelligence Research Society (FLAIRS'08)*, pp. 312-317, 2008.

[13] Su, X., Khoshgoftaar, T.M., Zhu, X., and Greiner, R. Imputation-Boosted Collaborative Filtering Using Machine Learning Classifiers, *the 23rd Annual ACM Symposium on Applied Computing (SAC'08)*, pp. 949-950,, 2008.