Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

# A Novel Bayesian Similarity Measure for Recommender Systems

**Guibing Guo, Jie Zhang, Neil Yorke-Smith***
Nanyang Technological University, Singapore
*American University of Beirut, Lebanon, and University of Cambridge, UK
{gguo1,zhangj}@ntu.edu.sg, nysmith@aub.edu.lb

## Abstract

*Collaborative filtering*, a widely-used user-centric recommendation technique, predicts an item's rating by aggregating its ratings from similar users. User similarity is usually calculated by *cosine similarity* or *Pearson correlation coefficient*. However, both of them consider only the direction of rating vectors, and suffer from a range of drawbacks. To solve these issues, we propose a novel Bayesian similarity measure based on the Dirichlet distribution, taking into consideration both the direction and length of rating vectors. Further, our principled method reduces correlation due to chance. Experimental results on six real-world data sets show that our method achieves superior accuracy.

## 1 Introduction

*Collaborative filtering* (CF) is one of the most widely-used user-centric recommendation techniques in practice [Zheng *et al.*, 2010]. The intuition is that users with similar preferences will have similar opinions (ratings) on new items. Similarity plays an important role. First, it serves as a criterion to select a group of similar users whose ratings will be aggregated as a basis of recommendations. Second, it is also used to weigh the ratings so that more similar users will have greater impact on the recommendations. Hence, similarity computation has direct and significant influence on the performance of CF. It is widely applied in both memory-based [Guo *et al.*, 2012] and model-based [Ma *et al.*, 2011] CF approaches.

The methods historically adopted to calculate user similarity in CF are cosine similarity (COS) and Pearson correlation coefficient (PCC) [Breese *et al.*, 1998]. COS defines user similarity as the cosine value of the angle between two vectors of ratings (the *rating profiles*); PCC defines user similarity as the linear correlation between the two profiles. It is well recognized that PCC and COS only consider the direction of rating vectors but ignore their length [Ma *et al.*, 2007]. Ahn points out that the computed similarity could even be misleading if vector length is ignored. PCC and COS are also known to suffer from several inherent drawbacks [Ahn, 2008]. These drawbacks can be summarized in four specific cases: (1) Flat-value problem: if all the rating values are flat, e.g., $(1, 1, 1)$, PCC is not computable as the correlation formula denominator becomes $0$, and COS is always $1$ regardless of the rating values; (2) Opposite-value problem: if two users specify totally opposite ratings on the commonly-rated items, PCC is always $-1$; (3) Single-value problem: if two users have only rated one item in common, PCC is not computable, and COS results in $1$ regardless of the rating values; (4) Cross-value problem: if two users have only rated two items in common, PCC is always $-1$ when the vectors cross, e.g., $(1, 3)$ and $(2, 1)$; otherwise PCC is $1$ if computable.

To address the above issues and propose a better similarity measure, we design a novel Bayesian approach by taking into account both the direction and length of rating vectors. An attractive advantage of Bayesian approaches is that one can infer in the same manner from a small sample as from a large sample [O'Hagan, 2004]. This is especially useful when the length of rating vectors is short. We apply the Dirichlet distribution to accommodate the multi-level distances between two ratings towards the same item (*rating pair*). Similarity is defined as the inverse normalization of user distance, which is computed by the weighted average of rating distances and of importance weights corresponding to the amount of rating pairs falling in that distance. We further exclude the probability of the scenario where users happen to be 'similar' due to a small number of co-rated items, termed as *chance correlation*. Experimental results based on six real-world data sets show that our approach can achieve superior accuracy.

## 2 Related Work

The 'traditional approaches' of PCC and COS are the most adopted similarity measures in the literature. Although it is reported that PCC works better than COS in CF [Breese *et al.*, 1998]—as the former performs data standardization whereas the latter does not—others show that COS rivals or outperforms PCC in some scenarios [Lathia *et al.*, 2008]. However, the literature rarely has sought to investigate the reasons for such phenomena, rather simply attributing them to the difference of data sets. We provide a reasonable and insightful explanation by conducting an empirical study on the nature of PCC, COS, and our method in Sections 3.2 and 3.3.

Various similarity measures have been proposed in the literature, given the ineffectiveness of the traditional approaches [Lathia *et al.*, 2008]. Broadly, they can be classified into two categories. First, some researchers attempt to modify the traditional measures in some way. Ma *et al.* [2007] propose a significance weight factor $\min(n, \gamma)/\gamma$ to devalue the PCC value when the number $n$ of co-rated items is small, where $\gamma$ is a constant and generally determined empirically. Shi *et al.* [2009] categorize users into different pools according to their preferences of items and then compute PCC similarity for each pool. However, these approaches do not make any changes to the calculation of PCC itself, and hence the inherent issues are not addressed.

Second, other researchers propose new similarity measures to substitute the traditional ones. Shardanand and Maes [1995] propose a measure based on the mean square difference (MSD) normalized by the number of commonly rated items. However, as we will show in Section 4, its performance is worse than PCC or COS. Lathia *et al.* [2007] develop a concordance-based measure which estimates the correlation based on the number of concordant, discordant and tied pairs of common ratings. It finds the proportion of agreement between two users. Since it depends on the mean of ratings to determine the concordance, this approach also suffers from the flat-value and single-value problems where user similarity is not computable. Ahn [2008] proposes the PIP measure based on three semantic heuristics: *Proximity*, *Impact* and *Popularity*. PIP attempts to enlarge the discrepancies of similarity between users with semantic agreements and those with semantic disagreements in ratings. However, the computed similarity is not bounded and often greater than 1, resulting in less meaningful user correlation. Bobadilla *et al.* [2012] propose the *singularities measure* (SM) based on the intuition that users with close ratings different from the majority (high singularity) are more similar than those with close ratings consistent with the others (low singularity). Although SM considers the mean of agreements, the length of rating vector is not taken into consideration. It tends to treat users with similar opinions as un-correlated if all of their ratings are consistent with others'. SM is evaluated only on a single data set in comparison with traditional approaches.

## 3 Bayesian Similarity

The proposed Bayesian similarity measure is distinct from PCC and COS, and aims to solve the issues of these traditional similarity measures. It takes into consideration both the direction (rating distances) and the length (rating amount) of rating vectors. Specifically, the rating distances are modelled by the Dirichlet distribution based on the amount of observed evidences, each of which is a pair of ratings (from the two vectors) towards a commonly rated item. Then the overall user similarity is modelled as the weighted average of rating distances according to their importance weights, corresponding to the amount of new evidences falling in the distance. Further, we consider the scenario where users happen to be 'similar' due to the small length of rating vectors, termed as *chance correlation*. Therefore, the length of rating vectors is taken into account via (1) the modelling of Dirichlet distribution, and (2) the chance correlation in our approach.

### 3.1 Dirichlet-based Measure

The *Dirichlet distribution* represents an unknown event by a prior distribution on the basis of initial beliefs [Russell and Norvig, 2009]. As more evidences come in, the beliefs of the event can be represented and updated by a posterior distribution. The posterior distribution well suits the similarity measure since the similarity is updated based on the records of new ratings of commonly-rated items issued by two users.

We first mathematically model the similarity computation using the Dirichlet distribution. Let $(r_{u,k}, r_{v,k})$ be a pair of ratings (i.e., rating pair) reported by users $u$ and $v$ on item $k$. The rating values are drawn from a discrete set $\mathcal{L} = \{l_1, \ldots, l_n\}$ ($l_{j+1} > l_j, j \in [1, n]$) of *rating scales* defined by a recommender system, where $n$ is the number of rating scales. Thus the rating distance can be denoted as $d = |r_{u,k} - r_{v,k}|$. We use the rating distance rather than rating difference in order to ensure the symmetry of similarity measure, i.e., $s_{u,v} = s_{v,u}$, where $s_{u,v}$ denotes the similarity between users $u$ and $v$. Let $D$ be a discrete random variable representing the level of *rating distance* between two ratings in a rating pair. $D$ takes values in the set $\mathcal{D} = \{d_1, \ldots, d_n\}$ of the supported levels of rating distances, where $d_i = |l_{j+i-1} - l_j|, d_{i+1} > d_i$, and $i, j, i+j-1 \in [1, n]$. For example, $d_1$ is the distance between two identical rating scales $l_j$. Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be the probability distribution vector of $D$, i.e., $P(D = d_i) = x_i$, which satisfies the additivity requirement $\sum_{i=1}^{n} x_i = 1$. The probability density of the Dirichlet distribution for variables $\boldsymbol{x} = (x_1, \ldots, x_n)$ with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ is:

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \prod_{i=1}^{n} x_i^{\alpha_i - 1}, \tag{1}$$

where $x_1, \ldots, x_n \geq 0$, $\alpha_1, \ldots, \alpha_n > 0$ and $\alpha_0 = \sum_{i=1}^{n} \alpha_i$. The parameter $\alpha_i$ can be interpreted as the amount of *pseudo-observations* of the event in question, i.e., rating pairs that are observed before real events happen. Hence, $\alpha_0$ is the total amount of prior observations. It is important to set appropriate values for the parameters $\alpha_i$ as they will significantly influence the posterior probability.

Before observing any rating pairs, and without any prior knowledge to the contrary, we assume that ratings from two users are random and uncorrelated. There are $n^2$ pseudo-observations corresponding to all the possible combinations of rating scales. Thus, parameter $\alpha_i$ will be the number of pseudo observations located in distance level $d_i$. Let $p_j$ be the prior probabilities of rating scales $l_j$. Thus we set the values of parameters $\alpha_i$ as follows:

$$\alpha_i = \begin{cases} \sum_{j=1}^{n} n^2 p_j^2 & \text{if } i = 1; \\ 2 \sum_{j=1}^{n-i+1} n^2 p_j p_{j+i-1} & \text{if } 1 < i \leq n. \end{cases} \tag{2}$$

Observe that the case of distance level $d_1$ only occurs when both ratings in a rating pair are identical, i.e., $(l_j, l_j)$. For other distance levels $d_i, 1 < i \leq n$, two combinations $(l_j, l_{j+i-1})$ and $(l_{j+i-1}, l_j)$ could produce the same rating distance at that level. Rather than setting these uninformed uniform parameters $\alpha_i$, we tried to learn prior probability of rating distances from the training data. However, experimental results did not show any advantages in performance. One possible explanation is that learning the exact distribution of ratings from training set may give rise to certain overfitting.

New evidence for the Dirichlet distribution is often represented by a vector. Specifically, the rating pair $(r_{u,k}, r_{v,k})$ can be represented by a vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)$ where only $\gamma_i = 1$ (where $i$ is such that $d_i = |r_{u,k} - r_{v,k}|$) and the remaining entries equal zero. For example, a rating pair (5, 3) on a certain item can be represented as $\boldsymbol{\gamma} = (0, 0, 1, 0, 0)$ if the rating scales are integers from 1 to 5. However, not all evidences will be considered as equally useful for similarity computation. Instead, we posit that realistic user similarity can only be calculated based on the (reliable) items with consistent ratings, and using the (unreliable) items with inconsistent ratings is risky and may cause unexpected influence on similarity computation. The rating consistency is determined by two factors: (1) the standard deviation $\sigma_k$ of ratings on item $k$; and (2) the rating tendency of all users. First, generally, the value of $\sigma_k$ reflects the extent of inconsistency of user ratings on item $k$. We define the acceptable range of rating deviations by $c\sigma_k$, where $c$ is a scale constant that can be adapted for different data sets. Second, however, the value of $\sigma_k$ may be less meaningful if the ratings on all items are highly deviated, i.e., users tend to disagree with each other in general. In this case, we consider the distance between the mode $r_m$ and mean $r_\mu$ of ratings, i.e., $d_{m,\mu} = |r_m - r_\mu|$. Since the mode represents the most frequently occurred value, the distance $d_{m,\mu}$ reflects the tendency of all user ratings. The greater the value of $d_{m,\mu}$ is, the more deviated user ratings are indicated and the less meaningful $\sigma_k$ will be. When $d_{m,u} > 1$,[1] $\sigma_k$ is not meaningful at all. Hence, the *important evidences* will be those whose rating distance for reliable item $k$ is within a small range $c\sigma_k$, given that users achieve agreements in most cases.

We define the evidence weight of $\gamma_i$ as:

$$e_i = \begin{cases} 1 & \text{if } c\sigma_k = 0; \\ 1 - \frac{d_i}{c\sigma_k} & \text{if } 0 \le d_i < 2c\sigma_k; \\ -1 & \text{otherwise.} \end{cases} \tag{3}$$

Let $\sigma$ be the standard deviation of all ratings in a recommender system. We restrict the important evidences within a range $c\sigma$ no more than the minimal rating scale $l_1$, i.e., $c = l_1/\sigma$. In case that the distributions of user ratings are unknown or that users generally do not have consensus ratings, we may set $c = 0$ so as not to consider evidence weights.

Now the Dirichlet distribution can be updated based on the observations of new evidences. Specifically, for an observation of a vector $\gamma$, the posterior probability density distribution will be $p(\boldsymbol{x}|\boldsymbol{\alpha} + \boldsymbol{\gamma})$. This procedure can be conducted sequentially to update the posterior probability density distribution when new rating pairs come in. Upon observation of $N$ rating pairs $\boldsymbol{\gamma}^1, \ldots, \boldsymbol{\gamma}^N$, the latest posterior probability density function becomes $p(\boldsymbol{x}|\boldsymbol{\alpha} + \sum_{j=1}^N \boldsymbol{\gamma}^j)$. Hence, the expected value of the posterior probability variable $x_i$ equals

$$E(x_i|\alpha_i + \gamma_i^0) = \frac{\alpha_i + \gamma_i^0}{\alpha_0 + \gamma^0}, \tag{4}$$

where $\gamma_i^0 = \sum_{j=1}^N \gamma_i^j e_i^j$ and $\gamma^0 = \sum_{i=0}^N \gamma_i^0$. $\gamma_i^j$ represents the $i$-th component of the $j$-th observation $\gamma^j$ and hence $\gamma_i^0$ is the amount of evidences whose rating distance is $d_i$.

Based on the posterior probability of each rating distance, we define *user distance* as the weighted average of rating distances $d_i$ according to their importance weights $w_i$:

$$d_{u,v} = \frac{\sum_{i=1}^n w_i \cdot d_i}{\sum_{i=1}^n |w_i|}, \tag{5}$$

where $d_{u,v}$ denotes the distance between two users $u$ and $v$, and $w_i$ represents the importance of the rating distance $d_i$ for calculating the user distance. Intuitively, the more new evidences that are accumulated at a rating distance $d_i$, the more important the distance $d_i$ will be. Hence, the importance weight of $d_i$ is computed by:

$$w_i = E(x_i|\alpha_i + r_i^0) - E(x_i|\alpha_i), \tag{6}$$

where we constrain $w_i > 0$ in order to remove the situation where posterior probability is less than priori probability, which can arise when a rating level receives few evidences (relative to all evidences). Then, normalizing the distance:

$$s'_{u,v} = 1 - \frac{d_{u,v}}{d_n}, \tag{7}$$

where $s'_{u,v}$ denotes the 'raw' similarity between two users $u$ and $v$, and $d_n$ is the maximum rating distance.

Until now, we have defined user similarity according to the distributions of rating distances. However, it is possible that two users are regarded as similar just because their rating distances happen to be relatively small, especially when the number of ratings is small. Hence it would be useful to reduce such correlation due to chance. Of $\gamma^0$ evidences, $\gamma_i^0$ evidences locate at the level of distance $d_i$. Recall that the prior probability of rating pairs with rating distance $d_i$ is $\alpha_i/\alpha_0$, and so the chance that $\gamma_i^0$ evidences fall in that level independently will be $(\alpha_i/\alpha_0)^{\gamma_i^0}$. Hence, the chance correlation is computed as the probability that amount of evidences fall in different distance levels independently:

$$s''_{u,v} = \prod_{i=1}^n (\frac{\alpha_i}{\alpha_0})^{\gamma_i^0}. \tag{8}$$

Thus, the smaller $\gamma_i^0$ is, the larger $s''_{u,v}$ will be.

Another concern is that similarity measures usually possess a certain level of user bias, i.e., the estimated similarity tending to be higher or lower to some extent than the realistic one. We will elaborate this issue later in Section 3.3. Therefore, the user similarity can be derived by excluding the chance correlation and user bias from the overall similarity:

$$s_{u,v} = \max(s'_{u,v} - s''_{u,v} - \delta, 0), \tag{9}$$

where $s_{u,v}$ denotes the user similarity between users $u$ and $v$, and $\delta$ is a constant representing the general user bias. As analyzed in Section 3.3, our method will generally hold a limited user bias around 0.04, i.e., $\delta = 0.04$.

### 3.2 Examples

Earlier we summarized four specific problems that PCC and COS suffer from. Here we illustrate by examples the differences among the similarity values computed by our Bayesian

[1]The value 1 is empirically determined based on the analysis of specifications of data sets that we will use in Section 4.

Table 1: Examples of PCC, COS and BS similarity metrics

| Problem | Examples | | | PCC | COS | BS | BS-1 |
|---|---|---|---|---|---|---|---|
| | ID | Vector $\boldsymbol{u}$ | Vector $\boldsymbol{v}$ | | | | |
| Flat-value | $a_1$ | [1, 1, 1] | [1, 1, 1] | NaN | 1.0 | 0.952 | 0.96 |
| | $a_2$ | [1, 1, 1] | [2, 2, 2] | NaN | 1.0 | 0.677 | 0.71 |
| | $a_3$ | [1, 1, 1] | [5, 5, 5] | NaN | 1.0 | 0.0 | 0.0 |
| Opp.-value | $a_4$ | [1, 5, 1] | [5, 1, 5] | -1.0 | 0.404 | 0.0 | 0.0 |
| | $a_5$ | [2, 4, 4] | [4, 2, 2] | -1.0 | 0.816 | 0.446 | 0.46 |
| | $a_6$ | [2, 4, 4, 1] | [4, 2, 2, 5] | -1.0 | 0.681 | 0.334 | 0.335 |
| Single-value | $a_7$ | [1] | [1] | NaN | 1.0 | 0.76 | 0.96 |
| | $a_8$ | [1] | [2] | NaN | 1.0 | 0.39 | 0.71 |
| | $a_9$ | [1] | [5] | NaN | 1.0 | 0.0 | 0.0 |
| Cross-value | $a_{10}$ | [1, 5] | [5, 1] | -1.0 | 0.385 | 0.0 | 0.0 |
| | $a_{11}$ | [1, 3] | [4, 2] | -1.0 | 0.707 | 0.332 | 0.383 |
| | $a_{12}$ | [5, 1] | [5, 4] | 1.0 | 0.888 | 0.530 | 0.5616 |
| | $a_{13}$ | [4, 3] | [3, 1] | 1.0 | 0.949 | 0.485 | 0.5623 |

similarity (BS) measure and the others. We denote BS-1 as the variant of our method that does not remove chance correlation. The results are shown in Table 1. All ratings in the table are integers in the range $[1, 5]$. We assume all the ratings are randomly distributed, i.e., $p_j = 0.2$ for Equation 2.

It is observed that our method can solve the four problems of PCC and COS, and generate more realistic similarity measurements overall. Specifically, for the flat-value and single-value problems, PCC is non-computable and COS is always 1, whereas BS produces more reasonable similarities. In addition, BS generates higher similarity in $a_1, a_2$ than in $a_7, a_8$ respectively. Although the rating directions are the same, the former situations have more amount of information than the latter. However, BS-1 computes the same values in these cases where chance correlation is not considered. BS-1 tends to generate larger values than BS. The differences between BS and BS-1 could be large, especially when the length of rating vectors is short (e.g., $a_2, a_7, a_8, a_{12}, a_{13}$). Further, when the ratings are diametrically opposite ($a_3, a_4, a_9, a_{10}$), BS always gives 0 no matter how much information we have. However, COS continues to generate relatively high similarity; PCC may not be computable and hence these values are unreasonable. When the ratings are opposite but not extreme ($a_5, a_6, a_{11}$), PCC gives the extreme value $-1$ all the time and COS tends to produce high similarity, whereas the similarity calculated by BS is kept low. Finally, if the ratings are not crossing ($a_{12}, a_{13}$), PCC will yield 1 if computable and COS produces large values relative to BS even if some of the ratings are conflicting. Hence, these values are counter-intuitive and misleading, as pointed out by Ahn [2008]. In contrast, our method can produce more realistic measurements.

### 3.3 Similarity Trend Analysis

In this subsection, we further investigate the nature of the three similarity measures in a more general way. The trends of computed similarity values are analyzed when the length of rating vectors varies in a large range, using the same settings as previous subsection. In particular, a normal distribution is used to describe the distribution of user similarity. Since similarity value is located in $[0, 1]$, the mean value of user similarity will be equal to the median of the normal distribution, i.e., $0.5$. Note that for comparison purpose, PCC similarity is normalized from $[-1, 1]$ to $[0, 1]$ via $(1 + \text{PCC})/2$. We vary the length of rating vectors from 1 to 200. For each length, we randomly generate one million samples of two rating vectors and calculate the similarity for each pair by applying PCC, COS, and BS. The mean and standard deviation for each length are summarized and shown in Figure 1.

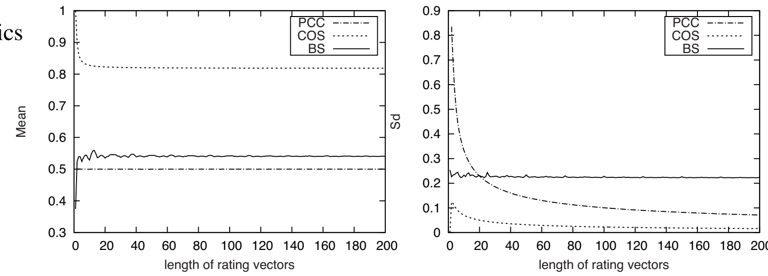
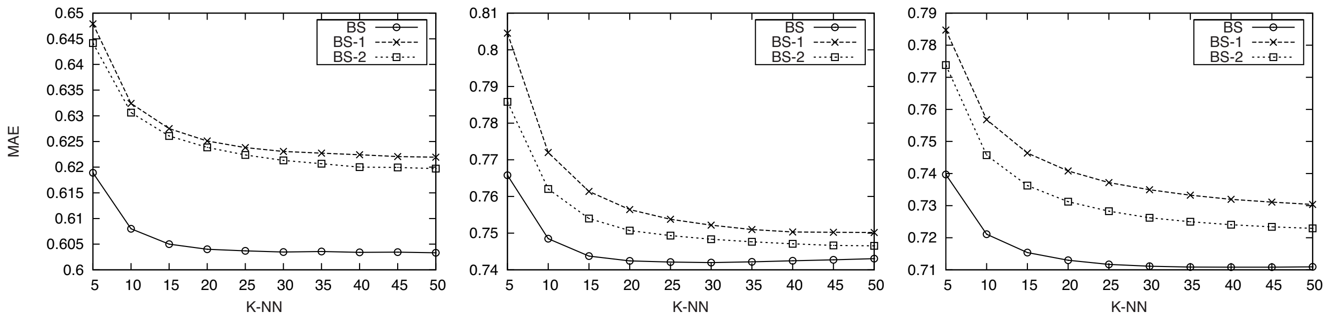



Figure 1: The trends of similarity measures

For the mean value, PCC stays at the value of 0.5, while COS starts with high values and decreases quickly (length $\leq$ 10), reaching a stable state with value of $0.82$. In contrast, BS begins with a low value at length 1 and then stays around $0.54$ with a limited fluctuation when the length is short. These results indicate that in general for any two users: (1) PCC is able to remove user bias; (2) COS always tends to generate high similarity around $0.82$, i.e., with a large bias around $0.32$; and (3) BS exhibits only a limited bias ($\delta = 0.04$). This phenomena is also observed by Lathia *et al.* [2008] who find that in the MovieLens data set (movielens.umn.edu), nearly $80\%$ of the whole community has COS similarity between $0.9$ and $1.0$, and that the most frequent PCC values are distributed around $0$ (without normalization), which corresponds to $0.5$ in our settings. For the standard deviation, PCC makes large deviations when the length of vectors is less than 20, COS generates very limited deviation, whereas BS keeps a stable deviation around $0.22$. In conclusion: (1) PCC is not stable and varies considerably when the vector length is short; (2) COS similarity is distributed densely around its mean value which makes it less distinguishable; and (3) BS tends to be distributed within a range of $0.22$ which makes its value more easily distinguishable from others.

## 4 Experiments

We evaluate recommendation performance using the 5-fold cross validation method. The data set is split into five disjoint sets; for each iteration, four folds are used as training data and one as a testing set. We apply the $K$-NN approach to select a group of similar users whose ranking is in the top $K$ according to similarity; we vary $K$ from 5 to 50 with step 5. The ratings of selected similar users are aggregated to predict items' ratings by a mean-centring approach [Desrosiers and Karypis, 2011]. Accuracy is measured by mean absolute error (MAE) between the prediction and the ground truth. Thus lower MAE indicates better accuracy. While our experiments use memory-based CF, we emphasize that similarity computation is equally relevant to model-based methods, including those based on matrix factorization such as Ma *et al.* [2011].




(a) FilmTrust (b) MovieLens 100K (c) MovieLens 1M

Figure 2: The effects of evidence weight and chance correlation

Table 2: Specifications of data sets in the experiments

| Data Set | # users | # items | # ratings | scales | $c$ |
|---|---|---|---|---|---|
| BookCrossing | 77.8K | 186K | 433K | [1, 10] | 0.5 |
| Epinions | 40.2K | 139.7K | 664.8K | [1, 5] | 0.0 |
| Flixster | 53.2K | 18.2K | 409.8K | [0.5, 5.0] | 0.0 |
| FilmTrust | 1508 | 2071 | 35.5K | [1, 5] | 0.6 |
| MovieLens 100K | 943 | 1682 | 100K | [1, 5] | 0.9 |
| MovieLens 1M | 6040 | 3952 | 1M | [1, 5] | 0.9 |

Table 3: Significance test results on all data sets

| Data Set | $t$ value | $p$ value | Best of Others | Alternative |
|---|---|---|---|---|
| FilmTrust | -7.0619 | 2.954e-05 | PCC | Less |
| MovieLens 1M | -4.4532 | 0.0007964 | SM | Less |
| BookCrossing | -40.3933 | 8.695e-12 | COS | Less |
| Flixster | -2.9545 | 0.008052 | SM | Less |
| MovieLens 100K | -0.9248 | 0.3792 | PIP | Two Sided |
| Epinions | 3.5688 | 0.003018 | SM | Greater |

**Data Sets.** Six real-world data sets are used in our experiments. Bookcrossing.com contains book ratings issued by users from the BookCrossing community. Epinions.com allows users to rate many items (books, movies, etc.) while Flixster.com is a movie rating and sharing community. FilmTrust (trust.mindswap.org/FilmTrust/) is also a movie sharing and rating website. Both MovieLens data sets (100K and 1M) are provided by the GroupLens group; each user has rated at least 20 items. The specifications of data sets are shown in Table 2, together with the computed values of $c$ (see Equation 3) in the last column.

## 4.1 Performance of BS and its Variants

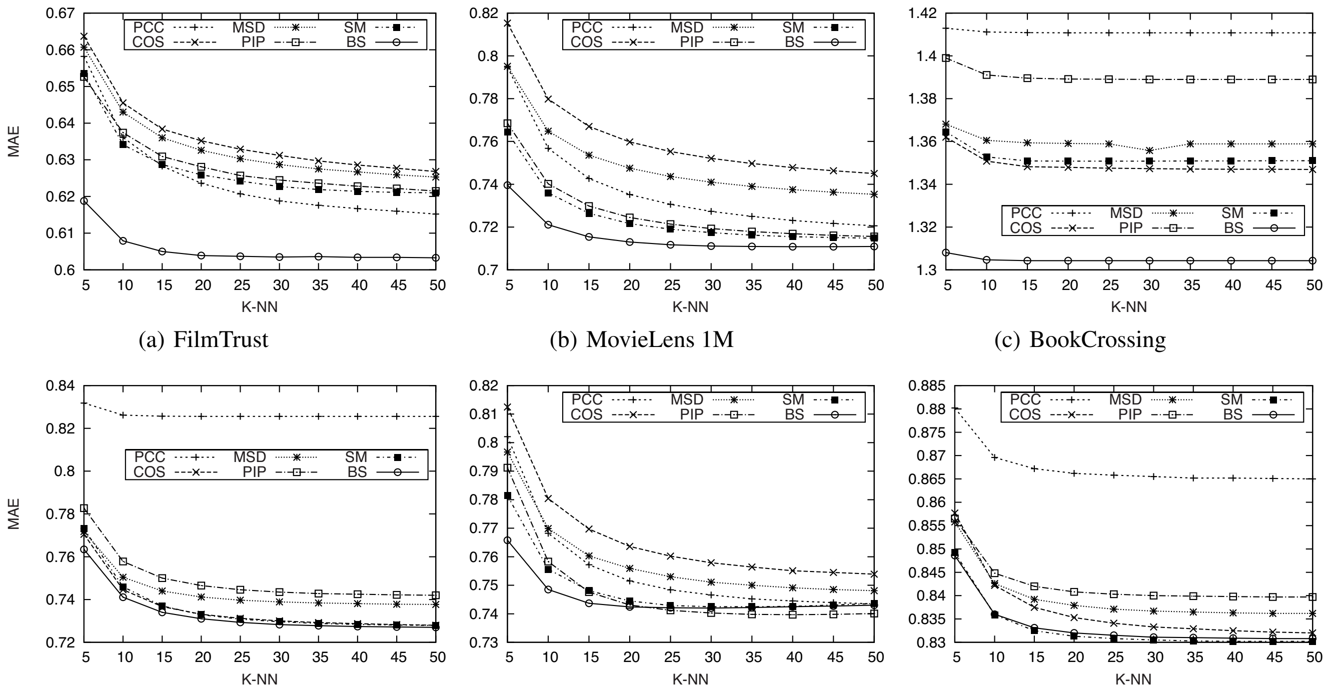
We first investigate the effects of two components in our approach BS, namely chance correlation and evidence weights. We denote BS-1 and BS-2 as the variants that disable chance correlation (setting $s''_{u,v} = 0$) and evidence weights (setting $c = 0$) from BS, respectively. The results on three data sets are illustrated in Figure 2 (and similar results occur in other data sets: graphs omitted for space reasons). It can be observed that BS consistently outperforms BS-2 which is in turn superior to BS-1, demonstrating the importance of both factors to our approach, and further indicating that disabling chance correlation will decrease the performance more than disabling the use of evidence weights. In other words, considering the length of rating vectors may have a greater impact on the predictive performance than other factors.

## 4.2 Comparison with other Measures

The baseline approaches are PCC, COS, and MSD. Besides these, we also compare with recent works, namely PIP and SM, as described in Section 2. The performance of these approaches is shown in Figure 3 in terms of MAE.

The results show that BS outperforms traditional measures (i.e., PCC and COS, also MSD) consistently in all data sets. Of the traditional measures, the performance of MSD is always between that of PCC and COS. PCC works better than COS in some cases (sub-figures a, b, e) and worse in others. One explanation is that PCC only removes local bias (the average of ratings on co-rated items) rather than global bias (the average of all ratings); hence it is not a standard data standardization. Of the newer methods, SM generally works better than PIP except for MovieLens 100K. One explanation is that PIP is especially designed for cold-start users whereas our experimental setting is for general users. Interestingly, PIP and SM outperform the traditional methods only in the two MovieLens data sets. This underscores the necessity of comparing performance in several different data sets. Adomavicius and Zhang [2012] also show that the accuracy of CF recommendations is highly influenced by the structural characteristics of data sets. By contrast, our method performs better than both PIP and SM in all data sets, except MovieLens 100K and Epinions, and exhibits greater improvements (with respect to traditional approaches). In MovieLens 100K, BS is still the best measure when $k$ is less than 25; after that, BS converges to the performance of SM which is slightly worse than the best performance achieved by PIP. In Epinions, BS and SM have very close performance and beat the others.

We conduct a series of paired two sample $t$-tests on all data sets to study the significance of accuracy improvement that our method achieves in comparison with the best of other methods (BOM) (confidence level 0.95). The results are shown in Table 3, where the types of *alternative* hypotheses are presented in the last column. The resultant $p$ values indicate that our method significantly ($p < 0.01$) outperforms all others for the first four data sets. For MovieLens 100K,




(a) FilmTrust (b) MovieLens 1M (c) BookCrossing

(d) Flixster (e) MovieLens 100K (f) Epinions

Figure 3: The predictive accuracy of comparative approaches

Table 4: Precision, Recall, F-measure on MovieLens 100K

| L | BS | SM | PIP | MSD | COS | PCC |
|---|---|---|---|---|---|---|
| 2 | **0.9801** | 0.9608 | 0.9653 | 0.9618 | 0.9602 | 0.9750 |
| | **0.4461** | 0.4365 | 0.4377 | 0.4365 | 0.4362 | 0.4426 |
| | **0.6131** | 0.6003 | 0.6023 | 0.6005 | 0.5999 | 0.6088 |
| 5 | **0.9580** | 0.9310 | 0.9453 | 0.9320 | 0.9286 | 0.9529 |
| | **0.5945** | 0.5805 | 0.5844 | 0.5805 | 0.5794 | 0.5903 |
| | **0.7337** | 0.7151 | 0.7223 | 0.7154 | 0.7136 | 0.7290 |
| 10 | **0.9119** | 0.8764 | 0.9049 | 0.8755 | 0.8709 | 0.9063 |
| | **0.6971** | 0.6787 | 0.6869 | 0.6785 | 0.6767 | 0.6921 |
| | **0.7902** | 0.7650 | 0.7810 | 0.7645 | 0.7616 | 0.7849 |
| 15 | **0.8706** | 0.8277 | 0.8609 | 0.8279 | 0.8211 | 0.8635 |
| | **0.7468** | 0.7251 | 0.7357 | 0.7249 | 0.7227 | 0.7410 |
| | **0.8040** | 0.7730 | 0.7934 | 0.7730 | 0.7688 | 0.7975 |
| 20 | **0.8338** | 0.7849 | 0.8216 | 0.7864 | 0.7777 | 0.8265 |
| | **0.7763** | 0.7521 | 0.7645 | 0.7521 | 0.7494 | 0.7701 |
| | **0.8040** | 0.7682 | 0.7920 | 0.7689 | 0.7633 | 0.7973 |

neither BS nor BOM are significantly better than the other. Only for Epinions is BS outperformed by another method (SM). Another set of significance tests show that our method achieves significantly better performance than the *second best* of other methods, i.e., SM ($p < 0.05$) and COS ($p < 0.01$) in MovieLens 100K and Epinions, respectively. Hence, looking across the range of data sets, we conclude that our method outperforms in general each other method considered.

Finally, to further explore MovieLens 100K, we look into the classification performance of all similarity methods in terms of precision, recall, and F-measure. We classify predictions greater than $4.5/5$ as relevant (to have a clear performance discrepancy) and otherwise as irrelevant. In Table 4, the first column (L) is the length of the recommended item list. The results confirm that BS consistently outperforms its counterparts on this data set as well.

## 5 Conclusion and Future Work

This paper proposed a novel Bayesian similarity measure for recommender systems based on the Dirichlet distribution, taking into account both the direction and length of rating vectors. In addition, correlation due to chance and user bias were removed to accurately measure users' correlation. Using typical examples, we showed that our Bayesian measure can address the issues of traditional similarity measures (i.e., PCC and COS). More generally, we empirically analyzed the trends of these measures and concluded that our method was expected to generate more realistic and distinguishable user similarity. The experimental results based on six real-world data sets further demonstrated the robust effectiveness of our method in improving the recommendation performance.

Our approach only relies on numerical ratings to model user correlation and hence it can be applied into many other domains, such as information retrieval. We plan to integrate more information about user ratings, such as the time when ratings were issued, in order to consider the dynamics of user interest [Li *et al.*, 2011], and to apply parameter learning for values $\delta$ and $c$ in our method.

**Acknowledgement.** Supported by the MoE AcRF Tier 2 Grant M4020110.020, and the Institute for Media Innovation at NTU.

## References


[Adomavicius and Zhang, 2012] G. Adomavicius and J. Zhang. Impact of data characteristics on recommender systems performance. *ACM Transactions on Management Information Systems (TMIS)*, 3(1):3, 2012.

[Ahn, 2008] H.J. Ahn. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, 178(1):37–51, 2008.

[Bobadilla *et al.*, 2012] J. Bobadilla, F. Ortega, and A. Hernando. A collaborative filtering similarity measure based on singularities. *Information Processing & Management*, 48(2):204–217, 2012.

[Breese *et al.*, 1998] J.S. Breese, D. Heckerman, C. Kadie, et al. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'98)*, pages 43–52, 1998.

[Desrosiers and Karypis, 2011] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. *Recommender Systems Handbook*, pages 107–144, 2011.

[Guo *et al.*, 2012] G. Guo, J. Zhang, and D. Thalmann. A simple but effective method to incorporate trusted neighbors in recommender systems. In *Proceedings of the 20th International Conference on User Modeling, Adaptation and Personalization (UMAP'12)*, 2012.

[Lathia *et al.*, 2007] N. Lathia, S. Hailes, and L. Capra. Private distributed collaborative filtering using estimated concordance measures. In *Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys'07)*, pages 1–8, 2007.

[Lathia *et al.*, 2008] N. Lathia, S. Hailes, and L. Capra. The effect of correlation coefficients on communities of recommenders. In *Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC'08)*, pages 2000–2005, 2008.

[Li *et al.*, 2011] B. Li, X. Zhu, R. Li, C. Zhang, X. Xue, and X. Wu. Cross-domain collaborative filtering over time. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, pages 2293–2298, 2011.

[Ma *et al.*, 2007] H. Ma, I. King, and M.R. Lyu. Effective missing data prediction for collaborative filtering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07)*, 2007.

[Ma *et al.*, 2011] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM'11)*, pages 287–296, 2011.

[O'Hagan, 2004] A. O'Hagan. Bayesian statistics: principles and benefits. *Frontis*, 3:31–45, 2004.

[Russell and Norvig, 2009] S.J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Prentice Hall, third edition, 2009.

[Shardanand and Maes, 1995] U. Shardanand and P. Maes. Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (SIGCHI'95)*, pages 210–217, 1995.

[Shi *et al.*, 2009] Y. Shi, M. Larson, and A. Hanjalic. Exploiting user similarity based on rated-item pools for improved user-based collaborative filtering. In *Proceedings of the 2009 ACM Conference on Recommender Systems (RecSys'09)*, pages 125–132, 2009.

[Zheng *et al.*, 2010] V.W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10)*, 2010.