A Novel Evidence-based Bayesian Similarity Measure for Recommender Systems

GUIBING GUO*, Northeastern University, China, and Nanyang Technological University, Singapore.
JIE ZHANG, School of Computer Engineering, Nanyang Technological University, Singapore.
NEIL YORKE-SMITH, American University of Beirut, Lebanon, and University of Cambridge, UK.

User-based collaborative filtering, a widely-used nearest neighbour-based recommendation technique, predicts an item's rating by aggregating its ratings from similar users. User similarity is traditionally calculated by *cosine similarity* or the *Pearson correlation coefficient*. However, both of these measures consider only the direction of rating vectors, and suffer from a range of drawbacks. To overcome 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. We posit that not all the rating pairs should be equally counted in order to accurately model user correlation. Three different evidence factors are designed to compute the weights of rating pairs. Further, our principled method reduces correlation due to chance and potential system bias. Experimental results on six real-world data sets show that our method achieves superior accuracy in comparison with other counterparts.

Categories and Subject Descriptors: H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information filtering*; H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision support*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Recommender systems, Bayesian similarity, Similarity measure, Dirichlet distribution

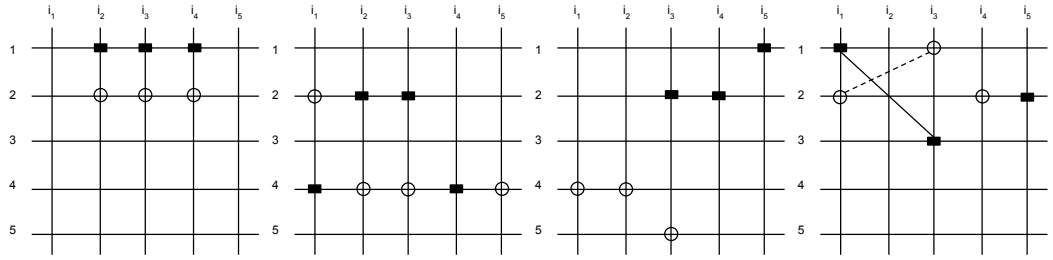
ACM Reference Format:
Guo, G., Zhang, J., Yorke-Smith, N., 2015. A Novel Evidence-based Bayesian Similarity Measure for Recommender Systems. *ACM Trans. Web* , , Article (February 2015), 30 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Recommender systems aim to provide users with personalized suggestions and thus ameliorate the *information overload* of an overwhelming number of choices of items. *User-based collaborative filtering* (CF) is one of the most widely-used nearest neighbour-based recommendation techniques in practice [Mohan et al. 2007; Cacheda et al. 2011]. The intuition is that users with similar preferences in the past are likely to have similar opinions (ratings) on new items in the future. Similarity plays an important role in CF techniques. 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

*Guibing Guo is the corresponding author.
A preliminary version of the work appeared at the IJCAI'13 conference [Guo et al. 2013], available via http://ijcai.org/papers13/Papers/IJCAI13-386.pdf.
Author's addresses: G. Guo, Northeastern University, China (current address), and Nanyang Technological University (where bulk research is done); email: guogb@swc.neu.edu.cn. J. Zhang, School of Computer Engineering, Nanyang Technological University, Singapore; email: zhangj@ntu.edu.sg. Neil Yorke-Smith, American University of Beirut, Lebanon, and University of Cambridge, UK., email: nysmith@aub.edu.lb.

DOI: http://dx.doi.org/10.1145/0000000.0000000



(a) Flat-value Problem (b) Opposite-value Problem (c) Single-value Problem (d) Cross-value Problem

Fig. 1. The problems of traditional similarity measures, i.e., Pearson correlation coefficient and cosine similarity. The filled rectangles and empty circles represent the ratings given by user u and user v, respectively.

used to weight 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 two main categories of CF techniques, namely memory-based [Guo et al. 2012; Ren et al. 2012] and model-based [Ma et al. 2011; Shi et al. 2013] approaches.

Cosine similarity (COS) and Pearson correlation coefficient (PCC) [Breese et al. 1998] are the methods most usually adopted to calculate user similarity in CF. 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 rating profiles. Formally, these 'traditional' similarity measures are defined by:

$$s_{u,v} = \frac{\sum_{i \in I_{u,v}} r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i \in I_{u,v}} r_{u,i}^2} \sqrt{\sum_{i \in I_{u,v}} r_{v,i}^2}} \quad \text{(COS)}$$

$$s_{u,v} = \frac{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{u,v}} (r_{v,i} - \bar{r}_v)^2}} \quad \text{(PCC)}$$

where $s_{u,v}$ is the similarity between user u and user v computed based on their ratings on the set $I_{u,v}$ of commonly rated items, $r_{u,i}$ denotes the rating given by user u on item i, and $\bar{r}_u$ and $\bar{r}_v$ represent the average rating given by user u and user v, respectively. Despite the popularity and simplicity of the two methods, it is well recognized that they only consider the direction of rating vectors but ignore the length [Ma et al. 2007]. Ahn [2008] points out that the computed similarity could even be misleading if vector length is ignored. Both PCC and COS are also known to suffer from several inherent drawbacks [Ahn 2008]. These drawbacks can be summarized in the following four specific cases, and illustrated in Figure 1 where two users u and v have given their ratings (from 1 to 5) on five items (denoted by $i_1, \ldots, i_5$). The ratings are represented by filled rectangles and by empty circles for user u and user v, respectively.

— **Flat-value problem**: if all the rating values are flat, e.g., $[1, 1, 1]$ given by user u on three items i_2, i_3, i_4 with values 1, or $[2, 2, 2]$ given by user v on the same items with values 2, PCC is not computable as the correlation formula denominator becomes 0, and COS is always 1 regardless of the rating values. In our example of Figure 1(a), since both users give low ratings to all the three items, their similarity should be computable (because of existing *rating pairs*[1]) and less than 1 (not exactly the same).

[1]A rating pair is defined as two ratings that are given by two users on a certain commonly rated item.

— **Opposite-value problem**: if two users specify opposite ratings on the commonly-rated items, PCC is always -1. As shown in Figure 1(b), user u disagrees with the ratings given by user v on the commonly rated items and the computed PCC is -1. However, two comments can be made: (1) the number of co-rated items has no effect on the computed PCC; and (2) their opinions are not extremely opposite in terms of rating semantics. Hence, the computed PCC value -1 could be misleading.
— **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. In Figure 1(c), although both users rated three items, only one item is commonly rated. In this case, PCC is not computable and COS yields 1. Both similarity measures cannot effectively reflect the situation where two users disagree with each other on the co-rated item.
— **Cross-value problem**: if two users have only rated two items in common, PCC is always -1 when the rating vectors cross each other, e.g., $[1, 3]$ and $[2, 1]$; otherwise PCC is 1 if computable. Both users in Figure 1(d) have rated two items in common and their ratings are crossing with each other. Then, PCC is computed as -1 indicating that they have distinct opinions about items i_1 and i_3. However, although the rating values are different and crossing, they both tend to give low ratings to these items and hence their opinions are similar to some extent.

To overcome the deficiencies of the traditional similarity measures, we propose a novel Bayesian similarity measure by taking into account both the direction and length of rating vectors. An attractive advantage of Bayesian approaches is that one can infer posterior probabilities 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 distance between two ratings in a 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. Three different evidence factors, namely rating consistency, Gaussian singularity and rating semantics, are developed to compute the weights of rating pairs. 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*, and remove the potential system bias caused by the formulation of Bayesian similarity. Experimental results based on six real-world data sets show that our approach achieves superior accuracy to other similarity measures.

This article extends and elaborates a preliminary version of the work that appeared at the IJCAI'13 conference [Guo et al. 2013]. The significant extensions are: (1) richer description, such as explanation of parameter settings, specifications of data sets, analysis of results, and survey of the literature; (2) technical improvements, namely besides the evidence weight factor, two more evidence factors are proposed and integrated with the previous factor to achieve better performance; (3) reworked experiments, which show that better performance is obtained, especially on the MovieLens data set where our method now outperforms all the others; (4) the effect of system bias is empirically studied; and (5) the performance of different similarity measures for cold-start users and niche items is investigated.

The rest of the article is organized as follows. Section 2 gives an overview of related studies on similarity measures. Our approach is elaborated in Section 3, including evidence weights, chance correlation, and system bias as the three main components. We exemplify the differences between traditional approaches and our proposal in Section 4.1, and conduct a more general study on the nature of those similarity measures in Section 4.2. Our approach is evaluated on six real-world data sets in Section 5. Finally, Section 6 concludes our current work and outlines the future research.

2. RELATED WORK

The 'traditional approaches' of PCC and COS are the most widely 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]. In other words, there is no consistent conclusion about the performance of PCC and COS in different cases. 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 Section 4.

Various similarity measures have been proposed in the literature, given the concerned issues 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. One direction is to weight computed similarity by taking into consideration the properties of commonly rated items. Breese et al. [1998] adopt the inverse user frequency as weights to restrict the contribution of popular items in the PCC computation. The intuition is that two users who agree on popular items are less likely to be similar than those who agree on less popular items, because users generally tend to like popular items. Said et al. [2012] design several weighting schemes based on the intuition that popular items have less impact on the similarity computation of PCC and COS than those receiving few number of ratings. The results show that the weighting schemes have only little effect on COS in all cases, while more discernible impact for PCC is only observable on some data sets for the users who rate many items. Breese et al. [1998] also propose to use a *case amplification* parameter ρ to transform the PCC value from w to w^ρ. A typical value of ρ in their experiments is 2.5. The transformation helps emphasize the high weights closer to 1, and suppress the low weights closer to zero. As a result, it can reduce noise in the data.

With the recognition of inability of PCC in cold conditions, 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 γ is a constant that is empirically determined. Similarly, Koren [2010] suggests $(n-1)/(n-1+\lambda)$ as a shrinking factor, where n is the number of co-rated items, and λ is a parameter determined by cross validation. Candillier et al. [2008] use Jaccard similarity as a weighting factor and combine it with other similarity measures (e.g., PCC) to appreciate the influence of co-rated items. All these weighting schemes can be regarded as the confidence of computed similarity. If similarity is based on sufficient ratings, we have a high confidence that the similarity is reliable and reflecting realistic user correlation; otherwise it has high chance that the similarity will be error-prone and even misleading.

Another direction is to enhance the similarity from the viewpoint of the properties of raters (i.e., users). Shi et al. [2009] categorize users into three different pools: 'positive', 'neutral' and 'negative' according to their rank preferences of items. Then, along with the similarity based on all ratings, three pool-based similarities are computed which will be integrated to produce recommendations. Ortega et al. [2013] adopt the concept of Pareto dominance to preprocess and narrow the whole user set to a set of users dominating others in terms of rating distances. Traditional similarity methods are then applied to compute user similarity. Although many approaches have been proposed, none of them makes any changes to the calculation of PCC or COS itself. As a result, however, the inherent issues mentioned in Section 1 are not addressed.

Second, instead of trying to modify traditional measures, other researchers propose *new* similarity measures. 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 5, its performance is generally worse than PCC or COS. Lathia et al. [2007] develop a concordance-based measure which estimates the user correlation based on the number of concordant, discordant and tied pairs of common ratings, i.e., 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*. The motivation is to explicitly consider the semantic meanings behind numerical rating values. For example, value 5 indicates a stronger preference than a value of 4, while both values mean that the user likes a specific item. Hence, the nuanced difference in semantics may matter and result in different similarity measurements. 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 mutually close ratings but inconsistent with the majority (high singularity) are more similar than those with close ratings and consistent with the others (low singularity). This measure bears similarity to the idea of popularity of items but differs in that singularity further investigates the deviation between one's rating and the majority's, and that even a popular item may receive distinct ratings from different users. 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 uncorrelated if all of their ratings are consistent with others'. SM was evaluated only on a single data set in comparison with traditional approaches. We evaluate it more thoroughly as part of our work.

Although these various approaches proposed to date have achieved improvements to some extent, two main criticisms can be suggested. First, a better similarity measure is expected to consistently perform better on different data sets. Second, most of these approaches are based on heuristics and lack a fundamental theoretical underpinning. We aim to develop a principled similarity measure that achieves a significant improvement in predictive accuracy when used in recommendations.

3. BAYESIAN SIMILARITY

In this section we present a novel similarity measure based on Bayesian inference, termed as *Bayesian similarity*. Section 3.1 introduces the model of Dirichlet distribution based on user ratings. Then, Section 3.2 elaborates three different factors to weight the importance of a rating pair, based on which a 'raw' user similarity is formulated in Section 3.3. Correlation due to chance and system bias are discussed in Sections 3.4 and 3.5, respectively. Finally, the pseudo-code of our algorithm and an illustrative example are presented in Section 3.6.

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 the rating vectors, termed as *chance correlation*. Therefore, the length of the rating vectors

is taken into account in our approach via (1) the modelling of Dirichlet distribution, and (2) the chance correlation.

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 new evidences come in, the beliefs of the event can be represented and updated by a posterior distribution. The posterior distribution well suits a similarity measure since the similarity is updated based on the records of new ratings of commonly-rated items issued by two users. In addition, many existing recommender systems are based on users' ratings on a number of discrete values (e.g., 1 to 5) which can be well handled by the Dirichlet distribution.

We first introduce a number of notations in order to mathematically model the similarity computation using the Dirichlet distribution. Let $(r_{u,k}, r_{v,k})$ be a pair of ratings (i.e., a rating pair) reported by users u and v on item k. The rating values are drawn from a discrete and equal-distance rating scale $\mathcal{L} = \{l_1, \ldots, l_n\}$ ($l_{j+1} > l_j, j \in [1, n-1]$) defined by a recommender system, where n is the number of different rating values. We define the *rating distance* as the absolute difference between two user ratings, i.e., $d = |r_{u,k} - r_{v,k}|$. 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. According to the rating scale $\mathcal{L}$, we can define $\mathcal{D} = \{d_1, \ldots, d_n\}$ as a set of possible rating distances, where $d_i = |l_{j+i-1} - l_j|, d_{i+1} > d_i$, for any $i, j, i+j-1 \in [1, n]$. For example, d_1 is the distance between two identical rating values l_j, i.e., $d_1 = 0$, and $d_n = l_n - l_1$ is the maximum rating distance between any two rating values.

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 given by:

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \prod_{i=1}^{n} x_i^{\alpha_i - 1}, \tag{2}$$

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 α_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, α_0 is the total amount of prior observations. It is important to set appropriate values for the parameters α_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 as illustrated in Table I. The rows and columns represent the first and second elements in the rating pairs whose values are taken from the predefined rating scale $\mathcal{L}$, and each entry is the rating distance of a corresponding rating pair. Therefore, there are n^2 pseudo-observations corresponding to all the possible combinations of rating values. Thus, parameter α_i will be the number of pseudo observations located in rating distance d_i.

Table I. The distribution of prior rating evidences

	$\boldsymbol{l_1}$	$\boldsymbol{l_2}$	$\ldots$	$\boldsymbol{l_{n-1}}$	$\boldsymbol{l_n}$
$\boldsymbol{l_1}$	d_1	d_2	$\ldots$	d_{n-1}	d_n
$\boldsymbol{l_2}$	d_2	d_1	$\ldots$	d_{n-2}	d_{n-1}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{l_{n-1}}$	d_{n-1}	d_{n-2}	$\ldots$	d_1	d_2
$\boldsymbol{l_n}$	d_n	d_{n-1}	$\ldots$	d_2	d_1

For generality, let p_j be the prior probability of a rating being value l_j taken out of the rating scale $\mathcal{L}$. Then we set the values of parameters α_i as follows:

$$\alpha_i = \begin{cases} \sum_{j=1}^{n} n^2 p_j^2 & \text{if } i = 1; \\ 2\sum_{j=1}^{n-i+1} n^2 p_j p_{j+i-1} & \text{if } 1 < i \leq n. \end{cases} \tag{3}$$

To explain this parameter setting, observe that the case of rating distance d_1 only occurs when both ratings in a rating pair are identical, i.e., (l_j, l_j), the probability of an identical rating pair with both rating values l_j is the multiplication of their respective probabilities, i.e., p_j^2. Hence, the estimated number of pseudo-observations (at the rating distance of d_1) is $n^2 p_j^2$, and thus the total number of such kind of pseudo-observations is the summation over all the possible rating values, i.e., $j \in [1, n]$. For the other distance levels $d_i, 1 < i \leq n$, two possible combinations (l_j, l_{j+i-1}) and (l_{j+i-1}, l_j) could produce the same rating distance, leading to the estimated number of pseudo-observations being $n^2(p_j p_{j+i-1} + p_{j+i-1} p_j) = 2n^2 p_j p_{j+i-1}$. We then iterate all the possible rating values to yield the total number of pseudo-observations of this kind.

Since the values of parameter α_i have important influence on the computation of posterior probability for the Dirichlet distribution, we proceed to determine the values of probabilities p_j (see Equation 3) for this purpose. In this article, we consider two possible ways, namely to learn from training data or to presume a simple uniform distribution. Experimental results show that the uninformed uniform parameters (i.e., $p_j = 1/n$) works as well as learning from the training data. One possible explanation is that learning the exact distribution of ratings from the training set may give rise to certain over-fitting. A deep analysis is necessary for further understanding. We leave it as a part of future work.

3.2. Evidence Weights

New evidence for the Dirichlet distribution is often represented by a vector. Specifically, we can represent the rating pair $(r_{u,k}, r_{v,k})$ by a vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)$ where $\gamma_i = 1$ (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 scale is the integers from 1 to 5. Such an evidence vector treats all evidences equally. However, in this article we claim that not all the evidences should and will be considered as equally useful for similarity computation. Three evidence factors are taken into account for this purpose.

3.2.1. Rating Consistency. The first factor we propose posits that realistic user similarity should be calculated based on the (reliable) items with consistent ratings, and that using the (unreliable) items with inconsistent ratings is risky and may cause unexpected influence on similarity computation. The motivation is because of the observation that most users tend to give positive ratings overall, for example, in Epinions[2] most users give rating values 4 and 5 (out of 5). Hence, it would be valuable to focus more on distinguishing the ratings on the consistent items.

Rating consistency is determined by two factors: (1) the standard deviation σ_k of ratings on item k; and (2) the rating tendency of all users. First, generally, the value of σ_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 σ_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_μ of

[2]Refer to Section 5.1 for the description of data sets used in our experiments.

ratings, i.e., $d_{m,\mu} = |r_m - r_\mu|$. Since the mode represents the most frequently occurring value, the distance $d_{m,\mu}$ reflects the tendency of all user ratings. The greater the value of $d_{m,\mu}$ is, the more user ratings are deviated and the less meaningful σ_k will be. When $d_{m,\mu} > 1$,[3] our experiments indicate that σ_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 γ_i as:

$$\varphi_k^i = \begin{cases} 1 & \text{if } c\sigma_k = 0; \\ 1 - \frac{d_i}{c\sigma_k} & \text{if } 0 \leq d_i < 2c\sigma_k; \\ -1 & \text{otherwise}, \end{cases} \tag{4}$$

where the upper bound $2c\sigma_k$ is chosen to restrict the value range to be $(-1, 1]$.

Let σ 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 value 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, or simply fall back to treating all evidences as equally important. The settings of c in different data sets used in our experiments are given in Table III. For BookCrossing, the mean and mode values are 7.6 and 8.0, respectively. Since $d_{m,\mu} = 0.4 \leq 1$, the value of c is given by $c = l_1/\sigma = 1.0/1.84 \approx 0.5$, where the standard deviation of all ratings is $\sigma = 1.84$. Therefore, for an item where the standard deviation of received ratings is σ_k, the smaller rating distance d_i is, the more important the evidence is. In contrast for Epinions, the mean and mode values are 3.99 and 5.0, and hence $d_{m,\mu} = 1.01 > 1$. In this case, we will set $c = 0$ to treat all evidences equally.

3.2.2. Gaussian Singularity. A commonly-used factor for similarity computation is called *singularity* [Bobadilla et al. 2012]. The intuition is that two users agree more if their ratings are close to one another but distinct from the majority, than they do if their ratings are close to the value of most users. Bobadilla et al. [2012] formulate singularity based on opinions, i.e., positive or negative ratings. Specifically, a rating that is greater than a certain rating value (often the median, e.g., 3 in the range [1, 5]) is regarded as a positive opinion; otherwise it is negative. They define the singularity of a positive (respectively, negative) rating as the proportion of negative (respectively, positive) opinions relative to the set of all opinions. In other words, a positive rating has high singularity if most ratings are negative.

However, this formulation ignores the differences between positive (or negative) opinions, that is, two ratings 4 and 5 are indifferently treated as positive opinions with the same singularity. Hence, we propose a more refined and general definition of singularity: the likelihood that a rating does not fall into the rating distribution. We use the assumption that a user's ratings given on all the items follow a Gaussian distribution, which can be adapted to both discrete and continuous rating scales. We term it *Gaussian singularity*.

Specifically, according to all the ratings of user u on item k, we can fit a Gaussian distribution $R \sim \mathcal{N}(\mu, \sigma)$, where μ and σ represent the average and standard deviation of user ratings. Then the singularity $\psi_{u,k}$ of a rating $r_{u,k}$ is computed using the probability density function as follows:

$$\psi_{u,k} = 1 - \frac{1}{\sqrt{2\pi}\sigma} exp\Big(- \frac{(r_{u,k} - \mu)^2}{2\sigma^2} \Big). \tag{5}$$

[3]The value 1 is empirically determined based on the analysis of specifications of six real-world data sets that we will use in Section 5.

Thus, the singularity of a pair of ratings $(r_{u,k}, r_{v,k})$ is computed by:

$$\psi^i_{u,v,k} = \psi_{u,k} * \psi_{v,k}, \tag{6}$$

where i refers to the subscript of rating evidence γ_i.

To illustrate, if two users u and v have each rated a number of items, we can use the ratings to fit two Gaussian probability distributions of ratings, respectively for each user. Suppose that for user u, the fitted distribution is $R_u \sim \mathcal{N}(4.2, 1.0)$, i.e., with the mean 4.2 and standard deviation 1.0. Then, the singularity of giving a rating $r_{u,k} = 5$ to item k is $\psi_{u,k} = 0.71$. Similarly for user v, given a fitted distribution $R_v \sim \mathcal{N}(3.0, 1.0)$, the singularity of rating item k as $r_{v,k} = 5$ is: $\psi_{v,k} = 0.95$. As expected, $\psi_{v,k}$ is greater than $\psi_{u,k}$ since it is more singular for user v to give a rating 5. Together, the overall singularity for two users giving both ratings as 5 (i.e., evidence γ_i) is: $\psi^i_{u,v,k} = 0.71 * 0.95 \approx 0.67$.

Although the Gaussian distribution is adopted in our work, it must be recognized that not all users' ratings follow exactly a Gaussian distribution. Other kinds of probability distributions may be used as an alternative. However, since we find that the Gaussian distribution produces good results (for most users), we will not work with alternative distributions here.

3.2.3. Rating Semantics. Ahn [2008] stresses the importance of considering the underlying semantics of rating scales in the computation of user similarity. Specifically, Ahn [2008] defines three semantic factors, namely *Proximity*, *Impact*, and *Popularity*. However, the formulation of these factors is not bounded (often greater than 1) and thus cannot be used as evidence weights in our method. Hence, we adapt and generalize the original definitions and give new formulations for each factor.

— **Proximity** reflects the difference of two ratings in terms of positive and/or negative opinions. For example, a pair of ratings (5, 3) is closer with each other than a pair of ratings (4, 2). Although the rating distance is the same, the former pair of ratings are both positive whereas the latter contains different opinions. Note that a rating that is less than the median of a rating scale is regarded as negative opinions; otherwise it is positive. Hence, we can define the agreement of two ratings as:

$$\text{agreement} = \begin{cases} \text{True} & \text{if } (r_{u,k} - r_{med})(r_{v,k} - r_{med}) \geq 0; \\ \text{False} & \text{otherwise}, \end{cases} \tag{7}$$

where r_{med} is the median rating of a rating scale predefined by a recommender system, given by $r_{med} = (l_1 + l_n)/2$. Then the proximity is defined by:

$$pr^i_{u,v,k} = \begin{cases} 1 - \dfrac{d_i}{d_n} & \text{if agreement is True}; \\ -\dfrac{d_i}{d_n} & \text{otherwise}, \end{cases} \tag{8}$$

where d_n is the maximal distance implied by a rating scale. Unlike the Gaussian Singularity focusing on the differences of specific rating values, the *Proximity* views user ratings from a more abstract level—the level of opinion. That is, both ratings 4 and 5 (out of 5) are regarding as the same positive opinions, but they differ in the level of singularity.

— **Impact** considers the extent to which an item is preferred or disliked by users. For example, a rating 1 (out of 5) means a user does not like an item at all while a rating 4 indicates a strong preference. To facilitate discussion, we denote $\mu = (r_{u,k} + r_{v,k})/2$ as the average rating of the pair. For a pair of ratings, we consider three cases:

(1) If both ratings are positive, the greater μ is, the more preferred the item will be.

(2) If both ratings are negative, the smaller μ is, the more disliked the item will be.
(3) If the opinions are different, the smaller μ is, the less distinct two opinions will be in terms of like and dislike.

Here a rating is regarded as positive if it is greater than or equal to the median of a rating scale, e.g., if $r_{u,k} \geq r_{med}$; otherwise it is negative. Based on these considerations, we obtain the following formulations of the impact factor:

$$im^i_{u,v,k} = \begin{cases} \frac{\mu}{l_n} & \text{if case 1;} \\ 1 - \frac{\mu}{l_n} & \text{if case 2;} \\ -\frac{\mu}{l_n} & \text{if case 3,} \end{cases} \tag{9}$$

where l_n is the maximal rating value predefined by a recommender system. Both cases 1 and 2 show positive impact on user similarity since users agree with each other, whereas case 3 has negative impact due to the disagreement in user opinions.

— **Popularity** is similar to singularity in that it gives bigger value to the ratings whose values are further away from the average rating of a specific item. For example, consider identical rating pairs (4, 5) for two items k and p: the proximity and impact measures for the two items will be the same. However, if the average rating of item k is 3 and that of item p is 4, then the first pair on item k should be more important since it reflects the similarity of two users better. We denote $\bar{r}_k$ as the average rating of a specific item, and $\bar{d}_k = |(r_{u,k} + r_{v,k})/2 - \bar{r}_k|$ as the distance between rating pair and the average. Hence, we compute the popularity as follows:

$$po^i_{u,v,k} = \begin{cases} \frac{\bar{d}_k}{d_n} & \text{if } (r_{u,k} - \bar{r}_k)(r_{v,k} - \bar{r}_k) \geq 0; \\ -\frac{\bar{d}_k}{d_n} & \text{otherwise.} \end{cases} \tag{10}$$

Having defined our three PIP factors, following Ahn [2008], the rating semantics is defined by multiplying them together:

$$\eta^i_{u,v,k} = pr^i_{u,v,k} * im^i_{u,v,k} * po^i_{u,v,k}. \tag{11}$$

3.2.4. Factor Integration. The proposed three evidence factors, namely rating consistency, Gaussian singularity, and rating semantics, reflect the different aspects of user ratings and rating pairs. Some of the factors can partially overlap, such as singularity and semantics. Hence, an effective combination of these three factors may bring the benefits and combat the drawbacks of each factor simultaneously. Specifically, for the sake of generality and simplicity, in this article the commonly-used linear combination is adopted as follows. For simplicity, we drop the dependency subscripts u, v, k of the three evidence factors and write:

$$e_i = \beta_1 * \varphi^i + \beta_2 * \psi^i + \beta_3 * \eta^i, \tag{12}$$

where e_i is the overall evidence weight of a rating pair; β_1, β_2 and β_3 indicate the relative importance of the factors, rating consistency, Gaussian singularity and rating semantics, respectively; they are constrained by $\beta_1 + \beta_2 + \beta_3 = 1$ and $\beta_1, \beta_2, \beta_3 \in [0, 1]$. Tuning the best settings of parameters β_1 and β_2 is typically done by cross validation. The linear combination (with two freedom degrees β_1, β_2) can greatly reduce the searching space (in the range of $[0, 1]$) than using an affine combination with three independent parameters each of which varies in the whole space of real values. Nevertheless, the noted overlapping between singularity and semantics may result in the

dominance of one factor over the other. Hence, together with rating consistency, the best settings can be achieved. We will elaborate in detail in Section 5.3.

3.3. Raw User Similarity

We are now in the position to explain how the Dirichlet distribution can be updated based on the observations of new evidences. Specifically, for an observation of a vector γ, 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 any new rating pairs are observed. 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 probability that a rating distance of new rating pair is d_i given the observed data will be equivalent to the expected value of the probability variable x_i:

$$p(\mathcal{D}=d_i|\gamma_0)=E(x_i|\gamma_0)=\frac{\alpha_i+\gamma_0^i}{\alpha_0+\pi}, \tag{13}$$

where $\gamma_0^i=\sum_{j=1}^{N}\gamma_i^j e_i^j$ and $\pi=\sum_{i=1}^{n}\gamma_0^i$. Note that γ_i^j represents the i-th component of the j-th observation γ^j and e_i^j denotes the evidence weights of the j-th observation given by Equation 12, and hence γ_0^i is the amount of accumulated 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:

$$d_{u,v}=\frac{\sum_{i=1}^{n}w_i\cdot d_i}{\sum_{i=1}^{n}w_i}, \tag{14}$$

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 according to the amount of cumulated evidence γ_0^i between users u and v. For simplicity, we neglect the symbols u,v for importance weights. 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:

$$w_i=\max\big(0,p(d_i|\pi)-p(d_i)\big)=\max\Big(0,\frac{\alpha_i+\gamma_0^i}{\alpha_0+\pi}-\frac{\alpha_i}{\alpha_0}\Big)=\max\Big(0,\frac{\alpha_0\gamma_0^i-\alpha_i\pi}{\alpha_0(\alpha_0+\pi)}\Big), \tag{15}$$

where we constrain $w_i\geq 0$ in order to remove the situation where a posterior probability is less than a prior probability, which can arise when a rating level receives very few evidences (relative to all the evidences). We then normalize the distance to derive user similarity:

$$s'_{u,v}=1-\frac{d_{u,v}}{d_n}, \tag{16}$$

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

3.4. Chance Correlation

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, or *chance correlation* for short. As described above, γ_0^i out of γ_0 evidences are located at the level of distance d_i. Recall that the prior probability of rating pairs with rating

distance d_i is α_i/α_0, and so the chance that γ_0^i evidences fall in that level independently will be $(\alpha_i/\alpha_0)^{\gamma_0^i}$. Hence, the chance correlation is computed as the probability that any amount of evidences falls in different rating distances independently:

$$s''_{u,v} = \prod_{i=1}^{n} (\frac{\alpha_i}{\alpha_0})^{\gamma_0^i}, \tag{17}$$

where $s''_{u,v}$ is the chance correlation between users u and v. Note that small values of γ_i^0 (i.e., few evidences) lead to large chance correlation while big values of γ_i^0 (i.e., many evidences) result in indiscernible chance correlation.

3.5. System Bias and Bayesian Similarity

The final consideration we treat is that similarity measures usually possess a certain level of *system bias*, i.e., the estimated similarity tends to be higher or lower to some extent than the realistic similarity. Intuitively, the system bias is partially due to the bias caused by the formulation of similarity measures. For example, PCC removes user averages when computing user similarity whereas COS does not. We will elaborate this issue later in Section 4.2. Therefore, user similarity is derived by excluding the chance correlation and system bias from the 'raw' similarity:

$$s_{u,v} = \max(s'_{u,v} - s''_{u,v} - \delta, 0), \tag{18}$$

where $s_{u,v}$ denotes the user similarity between users u and v, and δ is a constant representing the general system bias. As analyzed in Section 4.2, our method will generally hold a limited system bias around 0.04, i.e., $\delta = 0.04$, given that only rating consistency is used to compute evidence weights. However, if three evidence factors are effectively combined together, since they may complement with each other, the system bias could be ignorable, i.e., $\delta = 0$ as discussed in Section 5.

3.6. Algorithm and Example

The pseudo-code of the computation of Bayesian similarity for two users u and v is presented in Algorithm 1. The algorithm takes as input users u and v's ratings R_u and R_v and their rated items I_u and I_v, the prior probability of rating values p_i, combinational parameters β_1, β_2, and a number of pre-computed constants: items' standard deviations σ_k and users' average ratings μ. The computed Bayesian similarity is returned as output. The whole algorithm consists of two main parts. The first part computes evidence weights (lines 1–11). Specifically, we first compute the Dirichlet parameters α_i based on the input prior probability p_i by Equation 3 (line 1). A variable γ_0 is initialized as 0 to accumulate the total amount of new evidences (line 2). For each commonly rated item (line 3), we obtain a new rating pair (line 4) whereby the rating distance can be computed (line 5). Then, we proceed to compute the three evidence factors subsequently (lines 6–8) which will be combined to yield the evidence weight e_i by Equation 12 (line 9) and summed to variable γ_0 (line 10).

The second part of the algorithm computes the similarity measure. We declare two variables sum_d and sum_w (line 12) to accumulate the summation of weighted distances and importance weights, respectively. For each rating distance d_i (line 13), we compute its importance weight by Equation 15. After accumulating all the values in sum_d and sum_w (lines 16–17), we can compute user distance $d_{u,v}$ by Equation 14 (line 20) and thus the 'raw' user similarity by Equation 16 (line 21). Once the chance correlation is computed (line 22), the Bayesian similarity for the two users can be derived by Equation 18 (line 23), i.e., by removing the chance correlation and system bias from the 'raw' user similarity.

ALGORITHM 1: The Computation of Bayesian Similarity

Input : users u, v's ratings R_u, R_v and rated items I_u, I_v, rating prior probabilities p_i, parameters β_1, β_2, items' standard deviations σ_k, users' average ratings μ.
Output: Bayesian similarity between users u and v, i.e, $s_{u,v}$

1 compute Dirichlet parameters α_i by Equation 3;
2 set $\pi \leftarrow 0$;
3 **foreach** $k \in I_u \cap I_v$ **do**
4 obtain a rating pair of users u, v: $(r_{u,k}, r_{v,k})$;
5 choose rating distance $d_i \leftarrow |r_{u,k} - r_{v,k}|$;
6 compute rating consistency φ_k^i by Equation 4;
7 compute Gaussian singularity $\psi_{u,v,k}^i$ by Equation 6;
8 compute rating semantics $\eta_{u,v,k}^i$ by Equation 11;
9 combine three factors to obtain evidence weight e_i by Equation 12;
10 $\gamma_0^i \leftarrow \gamma_0^i + e_i$;
11 **end**
12 set $sum_d \leftarrow 0$, $sum_w \leftarrow 0$;
13 **foreach** $d_i \in D$ **do**
14 compute importance weight w_i of rating distance d_i by Equations 15;
15 **if** $w_i > 0$ **then**
16 $sum_d \leftarrow sum_d + w_i * d_i$;
17 $sum_w \leftarrow sum_w + |w_i|$;
18 **end**
19 **end**
20 compute user distance $d_{u,v}$ by Equation 14: $d_{u,v} = sum_d / sum_w$;
21 compute the 'raw' similarity $s'_{u,v}$ by Equation 16;
22 compute chance correlation $s''_{u,v}$ by Equation 17;
23 **return** Bayesian similarity $s_{u,v}$ by Equation 18;

Regarding the time complexity of Algorithm 1, the most time-consuming part is the **foreach** loop in lines 3–11. Specifically, for each iteration, the computational complexity for each step (e.g., Equation 4 in line 6) can be completed in $O(1)$. Hence, the overall time complexity is $O(m)$, where m is the average number of co-rated items of two users. In other words, our similarity measure is linear to the number of items commonly rated by the two users. Therefore, the time complexity of Bayesian similarity is of the same order of magnitude as PCC and COS. This is also confirmed in our experiments where no significant difference is observed in terms of computational cost.

Here we give an intuitive example to show the procedure of Algorithm 1 step by step. Suppose that two users u and v have rated four items in common, and their rating profiles are $[2, 4, 4, 1]$ and $[4, 2, 2, 5]$, respectively. First of all, we need to determine the values of parameters α_i by Equation 3. We use the uniform distribution and thus $p_j = 1/5 = 0.2$ if rating values vary from 1 to 5 (i.e., $n = 5$). Accordingly, we can obtain: $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5) = (5, 8, 6, 4, 2)$ and $\alpha_0 = \sum_{i=1}^{5} \alpha_i = 25$. Hence, the prior probability distribution will be $(p(d_1), p(d_2), p(d_3), p(d_4), p(d_5)) = (0.2, 0.32, 0.24, 0.16, 0.08)$. For simplicity, we set $\beta_1 = 1, \beta_2 = \beta_3 = 0$, i.e., only the factor rating consistency is considered. Since the characteristics of the whole data set is unknown, according to Equation 4, the parameter c is set to 0 and the evidence weight $e_i = \varphi^i = 1$ for each rating pair. After collecting $\gamma^0 = 4$ new rating pairs, the posterior probability distribution turns to be $\big(p(d_1|\gamma_0), p(d_2|\gamma_0), p(d_3|\gamma_0), p(d_4|\gamma_0), p(d_5|\gamma_0)\big) = (5/29, 8/29, 9/29, 4/29, 3/29)$. Hence, the importance weights can be computed by Equation 15: $(w_1, w_2, w_3, w_4, w_5) = (-20/725, -32/725, 51/725, -16/725, 17/725)$. Using Equation 14, the user distance $d_{u,v}$ is obtained by: $d_{u,v} = (2 * 51/725 + 4 * 17/725)/(51/725 + 17/725) = 2.5$. Hence, the 'raw'

similarity is derived by $s'_{u,v} = 1 - d_{u,v}/d_n = 1 - 2.5/4 = 0.375$. Then, the chance correlation by Equation 17 is given by: $s''_{u,v} = 0.2^0 * 0.32^0 * 0.24^3 * 0.16^0 * 0.08^1 = 0.00110592$, and the system bias is taken as 0.04. Overall, the Bayesian similarity is determined by: $s_{u,v} = \max(0.375 - 0.00110592 - 0.04, 0) = 0.33389408 \approx 0.334$. This example is also presented as an instance (a_6) in Table II, where the computed PCC is -1 and COS value is 0.681.

4. SIMILARITY MEASURES ANALYSIS

This section aims to provide intuitive examples of different similarity measures in the light of the four specific issues summarized in Section 1, and to give insight into the nature of different similarity measures.

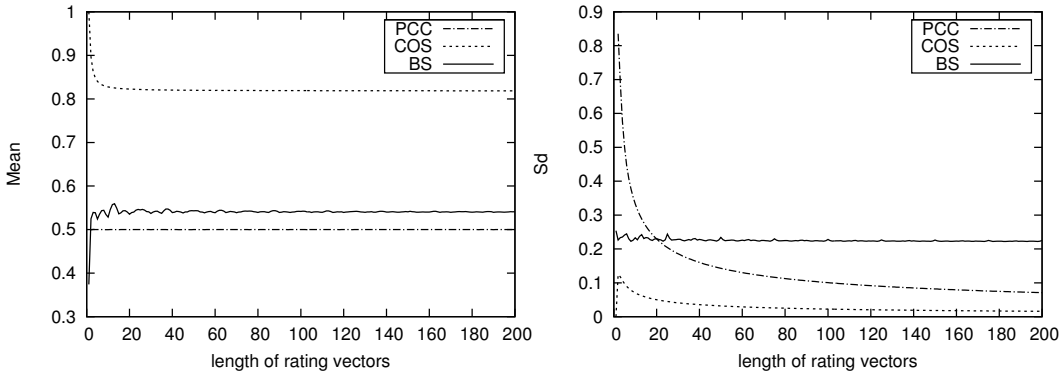
4.1. Examples

Earlier we summarized four specific problems from which PCC and COS suffer. Here we illustrate by examples the differences among the similarity values computed by our *Bayesian similarity* (BS) measure and the two traditional measures. We denote BS-1 as the variant of our method that does not remove chance correlation. The results are shown in Table II. All ratings in the table are integers in the range $[1, 5]$. We assume that the ratings are uniformly distributed, i.e., $p_j = 0.2$ for Equation 3. We only adopt the rating consistency to compute evidence weights, i.e., $\beta_1 = 1, \beta_2 = \beta_3 = 0$ for Equation 12, for simplicity and also partially due to the observation that rating consistency works better than other factors which will be analyzed in Section 5.3.

Table II. Examples of PCC, COS and BS similarity measures

Problems	**Examples**			**PCC**	**COS**	**BS**	**BS-1**
	ID	**Vector u**	**Vector 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
Opposite-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	0.336
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

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 a greater amount of rating evidences than the latter. Instead, BS-1 computes the same values in these cases where chance correlation is not considered. Overall, BS-1 tends to generate larger values than BS. The differences between BS and BS-1 are not trivial, especially when the length of rating vectors is short (e.g., $a_2, a_7, a_8, a_{12}, a_{13}$), which indicates the importance of removing chance correlation. 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. This behaviour matches intuition. However, COS continues to generate relatively high similarity while PCC may not be computable. When the ratings are opposite but not extreme (a_5, a_6, a_{11}), PCC gives the



(a) Mean (b) Standard Deviation

Fig. 2. The trends of similarity measures according to the variation of vector length

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 rating vectors 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.

4.2. Similarity Trend Analysis

We further investigate the nature of the three similarity measures in a more general manner. The trends of computed similarity values are analyzed when the length of rating vectors varies in a large range, using the same settings as in the previous subsection. In particular, a normal distribution is used to describe the distribution of user similarity. Since the 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 similarity measures. The mean and standard deviation for each length are summarized and shown in Figure 2.

For the mean value, PCC stays at the value of 0.5, while COS starts with high values and decreases quickly (length ≤ 10), reaching a stable state with the value of 0.82. Lin [1998] contends that one intuition a consistent similarity measure should obey is: the more commonality two users share, the more similar they are. In this regard, the COS similarity is counterintuitive in that it produces higher values when the length of rating vectors is short, and lower values when the length of rating vectors is long. In other words, the COS similarity is likely to be inconsistent when vector length is small. 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. Therefore, the BS similarity is more consistent than the COS similarity. These results indicate that in general for any two users: (1) PCC is able to remove system bias due to the data standardization involved; (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 ($\delta = 0.04$) under the experimental settings. This phenomenon is also observed by Lathia et al. [2008] who find that in the MovieLens data set, 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. A large deviation may cause the unstable values, i.e., inconsistent values are likely to be produced, while a small deviation may result in values that can not be well distinguished from each other. 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.

Note that our experiments assume that evidence weights are purely based on rating consistency. Under this condition, we find that our approach causes a limited system bias (0.04). As indicated by Equation 12, rating consistency can be combined with other evidence factors to form a more reliable and powerful factor to compute evidence weights. Thus, it is possible that the system bias can be further limited or eliminated by effectively combining the three evidence factors, and that the user similarity can be further distinguished by including more aspects of ratings. We will demonstrate the proposition in Section 5.4.

5. EVALUATION

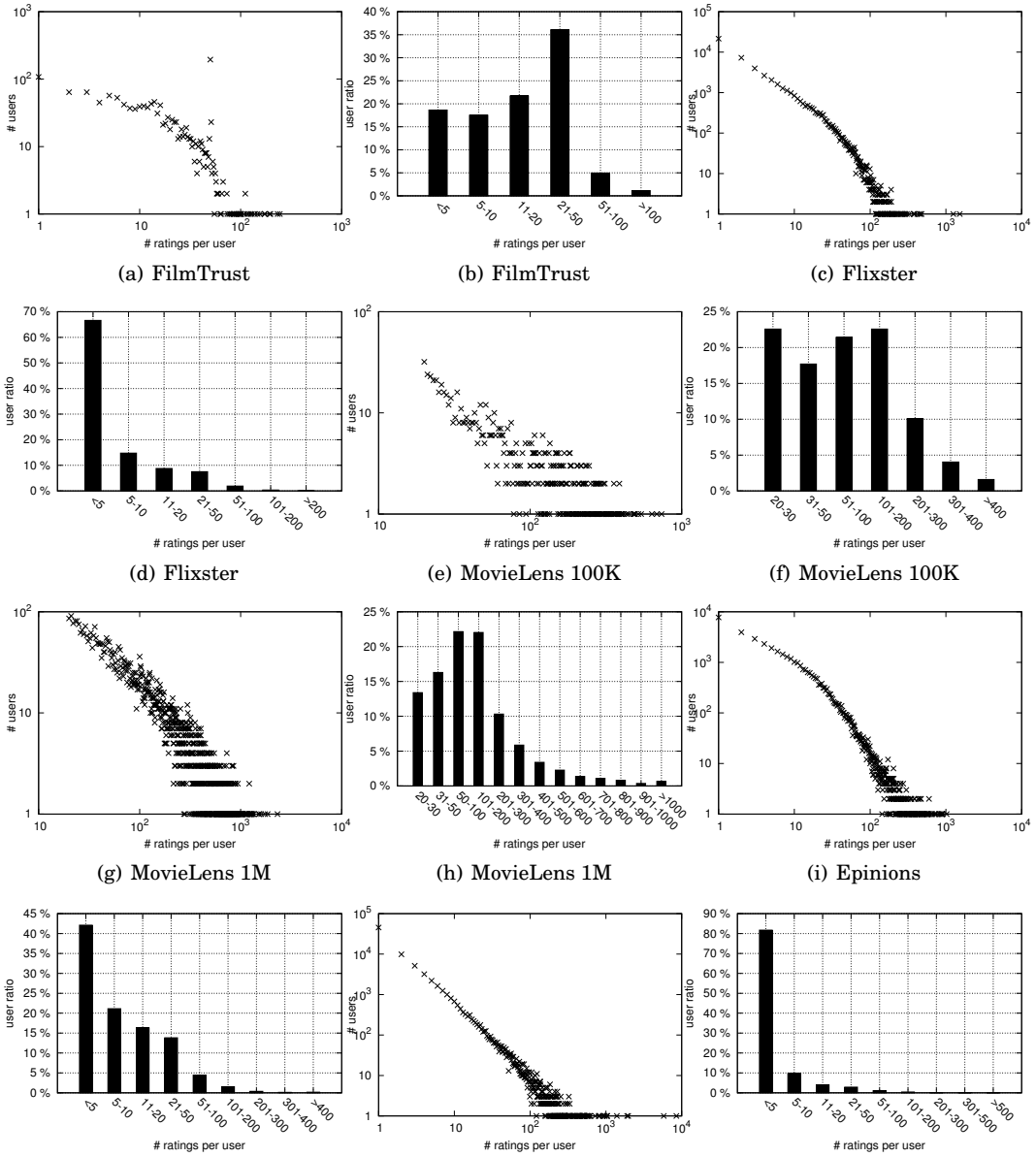
A series of experiments are conducted in this section to investigate: (1) the effects of different evidence factors as well as the best combinations of them on the performance of rating prediction; (2) the effects of chance correlation and system bias in our method; and (3) the performance of our approach in comparison with other similarity measures in terms of predictive accuracy.

5.1. Data Sets

Six real-world data sets are used in our experiments; their statistics are illustrated in Table III. They differ from each other in terms of predefined rating scales and density. BookCrossing.com is a free online book club to facilitate book sharing around the world. The data set[4] contains 433K ratings issued by 77.8K users on 186K books from the BookCrossing community. Epinions.com allows users to rate many different items (books, movies, etc.) by issuing an integer value from 1 to 5 and by adding textual review comments. The data set[5] includes 40.2K users, 139.7K items and 664.8K ratings. The remaining four data sets contain the data of three online communities in which users can give and share movie ratings with each other. Flixster[6] has the smallest rating density relative to the others and permits users to give more fine-grained and real-valued ratings from 0.5 to 5.0 with step 0.5. FilmTrust[7] is the smallest data set with only 35.5K user ratings. Notably, the two MovieLens data sets (100K and 1M)[8] have been pre-processed (by the GroupLens team) such that each user has rated at least 20 movies, resulting in the highest rating densities comparing with the others. The detailed specifications of all the data sets are presented in Table III, together with the computed values of c (see Equation 4) in the last column.

In addition, the distributions of the number of users with respect to the number of ratings given by per user are illustrated in Figure 3. The figure shows that generally

[4]http://www.informatik.uni-freiburg.de/~cziegler/BX/
[5]http://www.trustlet.org/wiki/Epinions_datasets
[6]http://www.cs.sfu.ca/~sja25/personal/datasets/
[7]http://www.librec.net/datasets/filmtrust.zip
[8]http://www.grouplens.org/node/12



(a) FilmTrust (b) FilmTrust (c) Flixster

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

(g) MovieLens 1M (h) MovieLens 1M (i) Epinions

(j) Epinions (k) BookCrossing (l) BookCrossing

Fig. 3. The distributions of the number of users with respect to the number of ratings given by per user across all the data sets

Table III. The statistics of data sets used in the experiments

Data Sets	# users	# items	# ratings	scales	density	c
FilmTrust	1508	2071	35.5K	[1, 5]	1.14%	0.6
Flixster	53.2K	18.2K	409.8K	[0.5, 5.0]	0.04%	0.0
MovieLens 100K	943	1682	100K	[1, 5]	6.30%	0.9
MovieLens 1M	6040	3952	1M	[1, 5]	4.47%	0.9
Epinions	40.2K	139.7K	664.8K	[1, 5]	0.05%	0.0
BookCrossing	77.8K	186K	433K	[1, 10]	0.03%	0.5

most users have only rated a small number of items (often no more than 20), and only a small portion of users have rated a large number of items. Since the MovieLens data sets have been pre-processed, each user has rated at least 20 items. On the other hand, different data sets show some distinct characteristics, for example, over 60% users have rated less than 5 items in Flixster and the ratio is even up to 80% in BookCrossing where the percentage is around 40% in Epinions and less than 20% in FilmTrust. The distributions on the other ranges of rating amounts also present the differences to some extent. In conclusion, the data sets vary from each other and thus represent a number of different kinds of communities with different types of users' rating patterns.

5.2. Experimental Settings

We evaluate recommendation performance using the 5-fold cross validation method. The data set is split into five disjoint subsets; for each iteration, four folds are used as training data and one as a test 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 in all the experiments. The ratings of selected similar users are aggregated to predict items' ratings by a mean-centering approach [Desrosiers and Karypis 2011]:

$$p_{u,i} = \bar{r}_u + \frac{\sum_{v \in N_u} s_{u,v}(r_{v,i} - \bar{r}_v)}{\sum_{v \in N_u} |s_{u,v}|}, \tag{19}$$

where $p_{u,i}$ is the predicted rating for user u on item i, N_u is the set of top K nearest neighbours, $s_{u,v}$ is the user similarity between users u and v, $\bar{r}_u$ and $\bar{r}_v$ are the average of ratings given by users u and v, respectively.

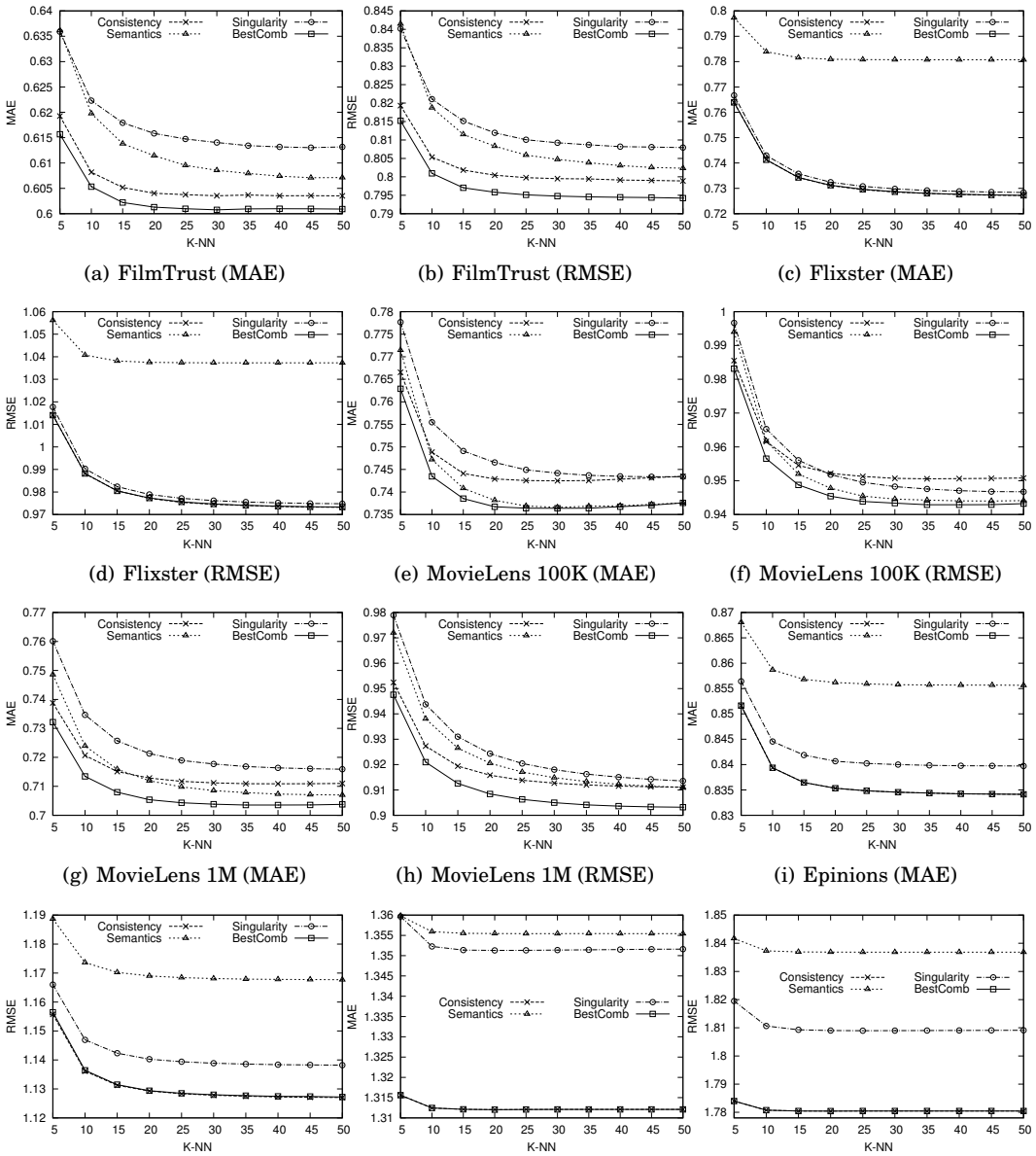
To study more aspects of the utilities of different similarity measures on recommendation performance, we consider three different testing views in our experiments.

— **All Users** is the view where all the ratings in the test set are used for prediction.
— **Cold Users** refers to the view where only the ratings of cold users (in the test set) who rated less than 5 items in the training set will be predicted.
— **Niche Items** refers to the view where only the ratings of niche items (in the test set) which received less than 5 ratings in the training set will be evaluated.

The predictive accuracy is measured by two popular metrics, namely mean absolute error (MAE) and root mean square error (RMSE) between the prediction $p_{u,i}$ and the ground truth $r_{u,i}$ using the test set:

$$\text{MAE} = \frac{\sum_{u,i \in \Omega} |p_{u,i} - r_{u,i}|}{|\Omega|}, \quad \text{RMSE} = \sqrt{\frac{\sum_{u,i \in \Omega} (p_{u,i} - r_{u,i})^2}{|\Omega|}} \tag{20}$$

where Ω represents the test set, and $|\Omega|$ is the cardinality of set Ω. Thus, lower MAE and RMSE values indicate better predictive 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] and Shi et al. [2013].



(a) FilmTrust (MAE) (b) FilmTrust (RMSE) (c) Flixster (MAE)

(d) Flixster (RMSE) (e) MovieLens 100K (MAE) (f) MovieLens 100K (RMSE)

(g) MovieLens 1M (MAE) (h) MovieLens 1M (RMSE) (i) Epinions (MAE)

(j) Epinions (RMSE) (k) BookCrossing (MAE) (l) BookCrossing (RMSE)

Fig. 4. The predictive performance using different evidence factors

5.3. Effects of Different Evidence Factors

Until now, we have introduced three different evidence factors to compute evidence weights, namely rating consistency, Gaussian singularity, and rating semantics. Hence it is necessary to investigate the impact of each evidence factor on the predictive performance as well as the best combination of three factors denoted by *BestComb*, obtained by tuning the values of parameters β_1 and β_2 in Equation 12. Specifically, we

conduct an exhaustive grid search[9] of the possible combinations of (β_1, β_2) and obtain their performance based on 5-fold cross validation while setting the number of most similar users as 10 for predictions[10], i.e., $K = 10$. The experiments show that the best combinations of parameters (β_1, β_2) are $(0.2, 0.1)$ on FilmTrust, $(0.8, 0.1)$ on Flixster, $(0.2, 0.2)$ on MovieLens 100K, $(0.1, 0)$ on MovieLens 1M, (0.9, 0.1) on Epinions and $(1, 0)$ on BookCrossing, respectively. After the grid search, we run 5-fold cross validation to show the predictive performance of different evidence factors. The results are illustrated in Figure 4. Significance tests (paired t-tests, confidence 0.95) are conducted between the best combination BestComb and the best of other single evidence factors. The significance test results are presented in Table IV.

In general, the best combination method, i.e., BestComb, achieves the best performance across all the data sets, and different individual evidence factors have various effect on different data sets. More specifically, Consistency reaches comparable results with BestComb on the Flixster, Epinions and BookCrossing data sets (i.e., in terms of MAE and RMSE, no significant differences). We note that the best combinational settings are $(0.8, 0.1)$, $(0.9, 0.1)$ and $(1, 0)$, respectively; where Consistency has the greatest influence on the overall combination. In addition, Consistency outperforms Singularity and Semantics on FilmTrust, and only performs worse than other single factors (i.e., Semantics) on two MovieLens data sets. Singularity performs the worst on FilmTrust, MovieLens 100K and 1M data sets, while Semantics is demonstrated to be the worst on the rest of the data sets. In conclusion, as a single evidence factor, Consistency is likely to be more reliable and effective than Semantics and Singularity.

As the best combination of three single factors, BestComb can always outperform the others over all the data sets, and significant improvements are observed on the FilmTrust, MovieLens 100K and 1M data sets (see Table IV). This may be explained by the fact that rating consistency focuses more on distinguishing similar ratings and that most users tend to give positive ratings, i.e., most ratings are likely to be similar to some extent. Recall in Section 4.2 we showed that our method can produce more realistic and distinguishable user similarities. In contrast, Gaussian singularity attends to consider more dissimilar ratings while rating semantics attempts to assume that ratings are distributed randomly. A proper integration of these evidence factors may benefit from each single factor and give the best predictive accuracy. Further, we note that rating consistency and semantics consistently have more important impact (i.e., greater coefficients) than Gaussian singularity across all the data sets. In other words, concentrating more on the similar ratings and taking into account their rating semantics can give the best value for similarity computation. One possible reason is that rating semantics has some overlapping with Gaussian singularity as explained in Section 3.2.4.

5.4. Effects of Chance Correlation and System Bias

After determining the best settings for parameters β_1 and β_2 in Equation 12, we proceed to explore the effects of the other two components of our approach BS, namely chance correlation and system bias. We denote BS-1 and BS-2 as the variants that disable chance correlation (setting $s''_{u,v} = 0$) and system bias (setting $\delta = 0$) in Equation 18, respectively. The experimental results are presented in Figure 5. It is observed that BS consistently outperforms BS-1 across all the data sets, though it is only slightly better than BS-1 on Flixster. Hence, we conclude that chance correlation is critical in our approach as disabling it will greatly decrease the predictive accuracy. However,

[9]The search space for each parameter is from 0 to 1 with step 0.1, with the constraint that $\beta_1 + \beta_2 \leq 1$.
[10]The setting $K = 10$ is chosen arbitrarily. In fact, other values of K also exhibit similar trends, indicating the suitability of our setting to investigate the effects of different evidence factors.

Table IV. Significance tests of the best combination *BestComb* w.r.t. the best of other single evidence factors in terms of MAE and RMSE across all the data sets, where p-values are denoted by the significance symbols: < 0.05 with *, < 0.01 with **, < 0.001 with ***; and 'NA' means not computable (or available).

Data Set (MAE)	**df**	t **value**	p **value**	**Best of Single Factors**
FilmTrust	9	-30.0271	1.232e-10***	Consistency
Flixster	9	4.1408	0.9987	Consistency
MovieLens 100K	9	-2.0664	0.03438*	Semantics
MovieLens 1M	9	-5.1098	3.183e-4***	Semantics
Epinions	9	NA	NA	Consistency
BookCrossing	9	NA	NA	Consistency
Data Set (RMSE)	**df**	t **value**	p **value**	**Best of Single Factors**
FilmTrust	9	-65.3828	1.156e-13***	Consistency
Flixster	9	3.5094	0.9967	Consistency
MovieLens 100K	9	-2.9583	8.002e-3**	Semantics
MovieLens 1M	9	-23.1675	1.237e-9***	Consistency
Epinions	9	4.6288	0.9994	Consistency
BookCrossing	9	NA	NA	Consistency

Table V. Significance tests of the best combination *BestComb* w.r.t. the best of other methods on the view of all users in terms of MAE and RMSE. Note that for the Epinions data set, two tests are available where the second one is with the *second* in MAE (*third* in RMSE) best method. The p-values are denoted by the significance symbols: < 0.05 with *, < 0.01 with **, < 0.001 with ***.

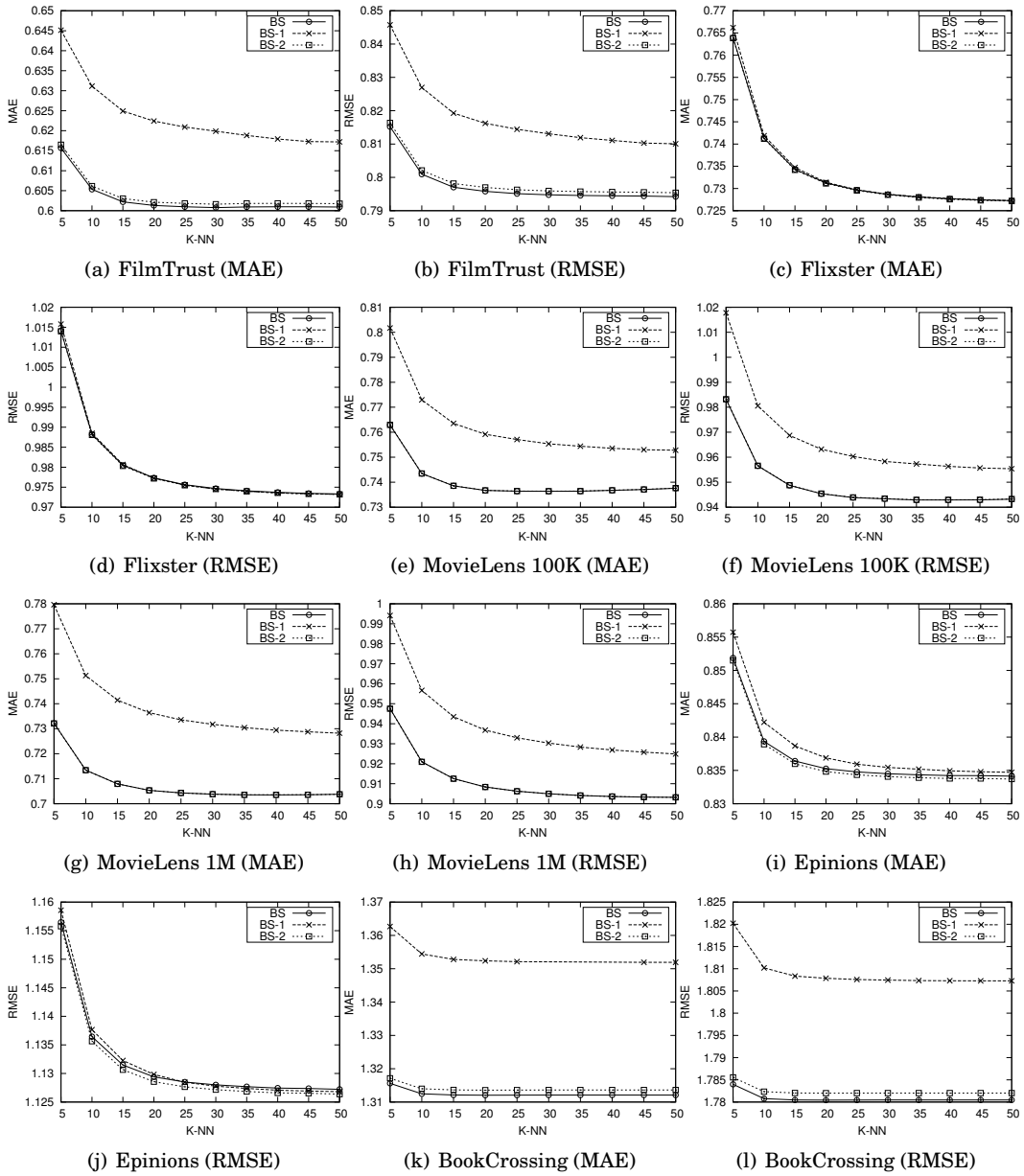
Data Set (MAE)	**df**	t **value**	p **value**	**Best of Other Methods**
FilmTrust	9	-7.4407	1.965e-5***	caPCC
MovieLens 1M	9	-7.9515	1.162e-5***	SM
BookCrossing	9	-32.6342	5.859e-11***	COS
Flixster	9	-2.7009	0.01218*	SM
MovieLens 100K	9	-3.0699	6.678e-3**	PIP
Epinions	9	3.4852	0.9966	SM
Epinions	9	-4.1937	1.164e-3**	COS
Data Set (RMSE)	**df**	t **value**	p **value**	**Best of Other Methods**
FilmTrust	9	-7.5298	1.790e-5***	caPCC
MovieLens 1M	9	-28.2894	2.096e-10***	SM
BookCrossing	9	-19.4044	5.926e-9***	COS
Flixster	9	-2.9437	8.194e-3**	SM
MovieLens 100K	9	-4.3736	8.939e-4***	PIP
Epinions	9	5.4422	0.9998	SM
Epinions	9	-34.0306	4.03e-11***	MSD

the effect of the system bias is not as significant as chance correlation. Specifically, BS-2 achieves comparable results with BS on most data sets and even exceeds BS on Epinions. That is, disabling system bias (i.e., $\delta = 0$) may cause only slight decrement or even sometimes reach slight increment relative to BS (with $\delta = 0.04$) in terms of predictive accuracy. As a conclusion, it is indiscernible in accuracy to disable system bias, though setting a small value (0.04) may result in slightly better performance. As explained in Section 4.2, the value 0.04 is obtained by using only the rating consistency to compute evidence weights. However, since a good combination usually requires the consideration of other evidence factors as demonstrated in previous subsection, it may lead to a more limited or ignorable system bias.

5.5. Performance Comparison on All Users

The baseline approaches for comparison are PCC, COS, MSD [Shardanand and Maes 1995], inverse user frequency-based COS (denoted by iufCOS) and case amplification-based PCC (denoted by caPCC) [Breese et al. 1998].[11] Breese et al. [1998] empirically

[11] Other variants of PCC exist in the literature, but we will not compare all of them in this work.



(a) FilmTrust (MAE) (b) FilmTrust (RMSE) (c) Flixster (MAE)

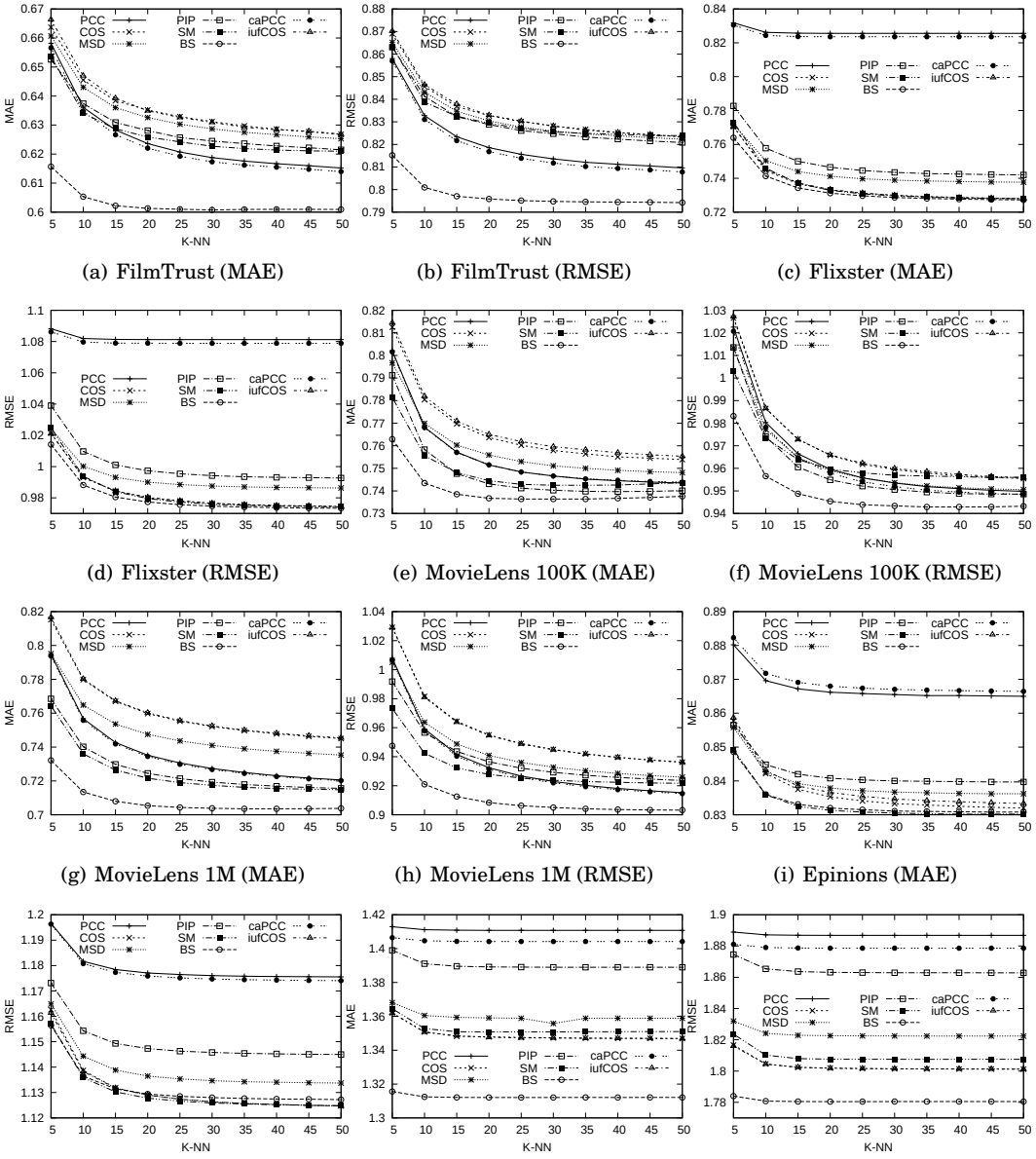
(d) Flixster (RMSE) (e) MovieLens 100K (MAE) (f) MovieLens 100K (RMSE)

(g) MovieLens 1M (MAE) (h) MovieLens 1M (RMSE) (i) Epinions (MAE)

(j) Epinions (RMSE) (k) BookCrossing (MAE) (l) BookCrossing (RMSE)

Fig. 5. The effects of disabling chance correlation or system bias

suggest that the best value of the case amplification parameter for caPCC is $\rho = 2.5$. We go further and adopt a grid search for each of our data sets in the value set $\{0.5, 1, 2, 2.5, 3, 5, 10\}$ to find out the optimal ρ values across all testing views. Experimental results show that the setting of $\rho = 0.5$ consistently achieves the best performance across all the test cases. Besides these five methods, we also compare with recent approaches, namely PIP [Ahn 2008] and SM [Bobadilla et al. 2012] which show better



(a) FilmTrust (MAE) (b) FilmTrust (RMSE) (c) Flixster (MAE)

(d) Flixster (RMSE) (e) MovieLens 100K (MAE) (f) MovieLens 100K (RMSE)

(g) MovieLens 1M (MAE) (h) MovieLens 1M (RMSE) (i) Epinions (MAE)

(j) Epinions (RMSE) (k) BookCrossing (MAE) (l) BookCrossing (RMSE)

Fig. 6. The predictive accuracy of comparative approaches on all users

performance than a number of baselines, as described in Section 2. The performance of these approaches is shown in Figure 6 in terms of MAE and RMSE.

The results show that BS outperforms traditional measures (i.e., PCC and COS, also MSD) consistently in all the data sets. Of the traditional measures, the performance of MSD is always between that of PCC and COS. PCC works better than COS on some data sets including FilmTrust, MovieLens 100K and 1M data sets and worse in the 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 the ratings); hence it is not a standard data standardization. With an optimized case amplification value (i.e., $\rho = 0.5$), caPCC slightly beats PCC consistently across all the data sets. On the other hand, with a discount of inverse user frequency, iufCOS works very closely to (or slightly worse than) COS, indicating the uselessness of weighting schemes for COS as reported by Said et al. [2012]. Of the newer methods, SM generally works better than PIP except on MovieLens 100K. Interestingly, PIP and SM outperform the traditional methods only on the two MovieLens data sets. This underscores the necessity of comparing performance on 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. In line with this conclusion, we observe that the performance of PIP varies on different data sets relative to other baselines. This may be explained by the grid formulation of the PIP method. For example, the factor of proximity [Ahn 2008] is set in such a way that the distance between two ratings will be doubled if they disagree with each other. Such a kind of setting may or may not work for some data sets, since it depends in part on the rating scale used in the data set. By contrast, our method performs better than both PIP and SM on all the data sets except Epinions, and exhibits greater improvements (with respect to the traditional approaches). On Epinions, BS and SM have very close performance and outperform the other methods.

In addition to the above experiments, we conduct a series of paired two sample t-tests on all the data sets to study the significance of accuracy improvement that our method achieves in comparison with the best of other methods (confidence level 0.95). The results are shown in Table V, where the *alternative* hypotheses are: *the MAE (RMSE) of BS is significantly less than that of the best of other methods*. The resultant p values indicate that our method significantly outperforms all others on five out of the six data sets. Only on Epinions is BS slightly outperformed by another method (SM). However, this performance difference on Epinions between BS and SM is quite small: 0.00047 in MAE and 0.00162 in RMSE on average. A further significance test is adopted to compare our method with the *second (or third) best* of other methods, i.e., COS in MAE (or MSD in RMSE), on Epinions. The results, also in Table V, show that BS achieves significantly better performance than the second/third best other method. Hence, looking across the range of data sets, we conclude that our method has the most robust performance of all the methods considered.

5.6. Performance Comparison on Cold Users

The performance on cold users is illustrated in Figure 7, and the corresponding significance tests are presented in Table VI with respect to the best of other methods. Since users in the two MovieLens data sets rated at least 20 items, these data sets are not suitable for the experiments by the definition of testing view of *Cold Users*.

A number of observations can be drawn from the experimental results. Firstly, in contrast with the performance on all users, the differences between PCC and caPCC is negligible on cold users. In other words, the weighting scheme for PCC on cold users is not helpful. By contrast, although the curves of COS and iufCOS are still highly overlapped, the difference is that iufCOS works better than COS when K is small. Generally, PCC works better than COS on FilmTrust and BookCrossing but worse on Flixster and Epinions.

Secondly and surprisingly, PIP achieves poor performance (and even worse than MSD) in general. This result seems to be in conflict with the conclusion reported by Ahn [2008]. One possible explanation is that the experimental settings in Ahn [2008] count the number of ratings used to calculate user similarity, whereas we focus on the number of ratings issued by the users. In this regard, our setting is more

Table VI. Significance test results on the view of cold users in terms of MAE and RMSE. The last test is between our method with the *second* best method in FilmTrust. The p-values are denoted by the significance symbols: < 0.05 with *, < 0.01 with **, < 0.001 with ***.

Data Set (MAE)	**df**	t **value**	p **value**	**Best of Other Methods**
FilmTrust	9	-13.0566	1.870e-7***	PCC
Flixster	9	-2.9292	8.389e-3**	SM
Epinions	9	-51.2374	1.032e-12***	SM
BookCrossing	9	-1680.479	<2.2e-16***	PCC
Data Set (RMSE)	**df**	t **value**	p **value**	**Best of Other Methods**
FilmTrust	9	19.2953	1.0	PCC
Flixster	9	-15.3618	4.585e-8***	SM
Epinions	9	-4.3494	9.259e-4***	SM
BookCrossing	9	-1495.982	<2.2e-16***	PCC
FilmTrust	9	-12.9557	1.999e-7***	iufCOS

realistic (for selecting cold users) since one interaction with other users does not mean that the user only rated one item; rather many items could have been rated. In contrast, SM works relatively better than the other baselines on all the data sets.

Lastly and most importantly, our approach BS in general works significantly better (see Table VI) than the others across all the data sets, except FilmTrust, in terms of RMSE. Specifically, as shown in the table regarding the performance in FilmTrust, our approach BS works better than the best of other methods in MAE, but worse than PCC in RMSE. The lower MAE value indicates that the rating predictions by BS are generally closer to the ground truth than PCC (see Figure 7a), while the higher RMSE value means that BS produces greater errors than PCC[12] (see Figure 7b). In other words, the rating predictions by BS tend to be either greatly approximated (mostly due to small MAE) or deviated (few due to relatively high RMSE) in FilmTrust. This can be attributed to the accuracy of computed factors. For example, in the case that a user has only a few ratings, the average of her ratings may vary more than the case where many ratings are available. This may lead to incorrect estimation of the factors such as impact (see Equation 9), popularity (see Equation 10), Gaussian singularity (see Equation 5), etc. Another explanation for the variance between MAE and RMSE is due to the small size of FilmTrust, which makes the performance more sensitive to a few number of high predictive errors. Nevertheless, from the last test presented in Table VI, the performance of our approach BS still performs significantly better than the second best of other methods.

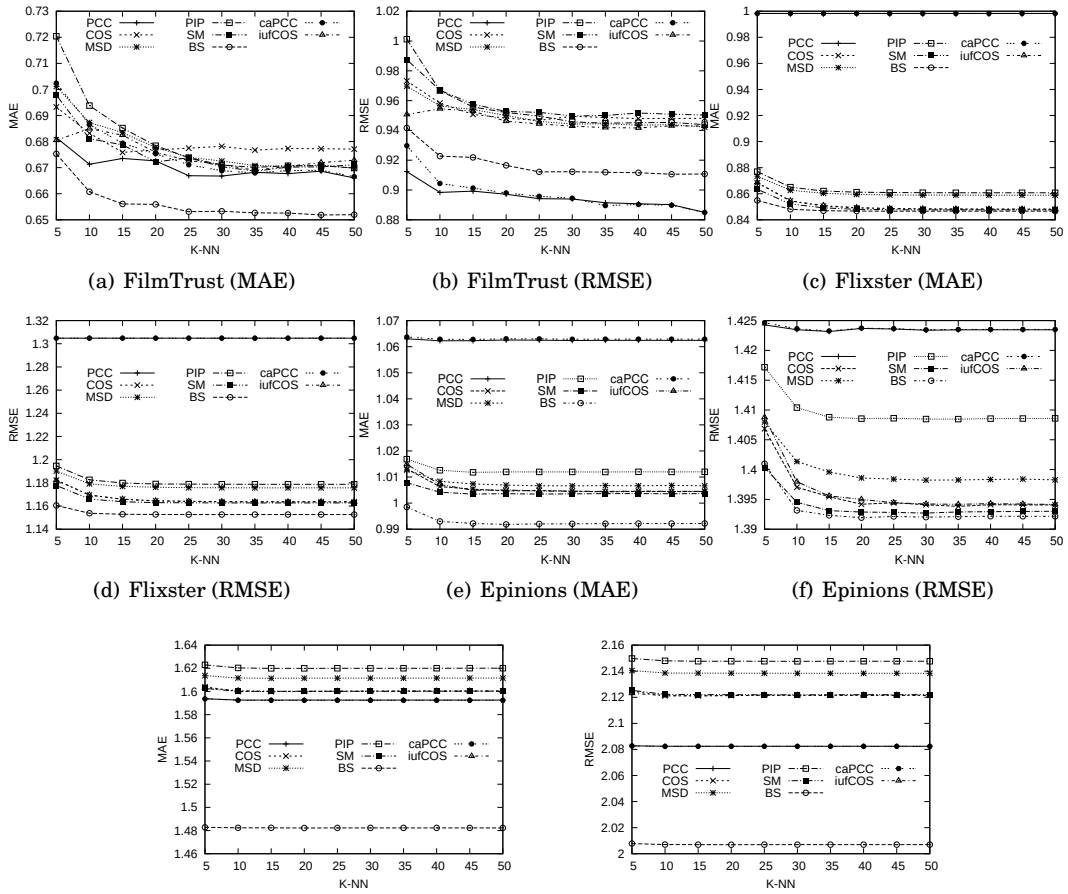
5.7. Performance Comparison on Niche Items

The performance of all methods on niche items is shown in Figure 8. By definition, niche items are those which received less than 5 ratings. Hence, there is no need to tune the number K of nearest neighbours for a specific user since the maximum number will be less than 5. Overall, the performance on niche items is similar to that on all users. Specifically, PCC is inferior to caPCC, while COS is similar to iufCOS in terms of predictive accuracy. PIP shows no better results than the other baselines, but differently SM tends to act similarly as the others including MSD, COS and iufCOS. It is noted that our approach consistently outperforms all the others across different data sets except Flixster and Epinions where BS performs close to or only slightly worse than the best of other methods.

5.8. Summary and Discussion

In summary, the empirical results show that our approach, BS, works better than the others in terms of both MAE and RMSE across a number of real-world data sets. Even

[12]By definition, RMSE gives relatively high weights on large predictive errors.



(a) FilmTrust (MAE) (b) FilmTrust (RMSE) (c) Flixster (MAE)

(d) Flixster (RMSE) (e) Epinions (MAE) (f) Epinions (RMSE)

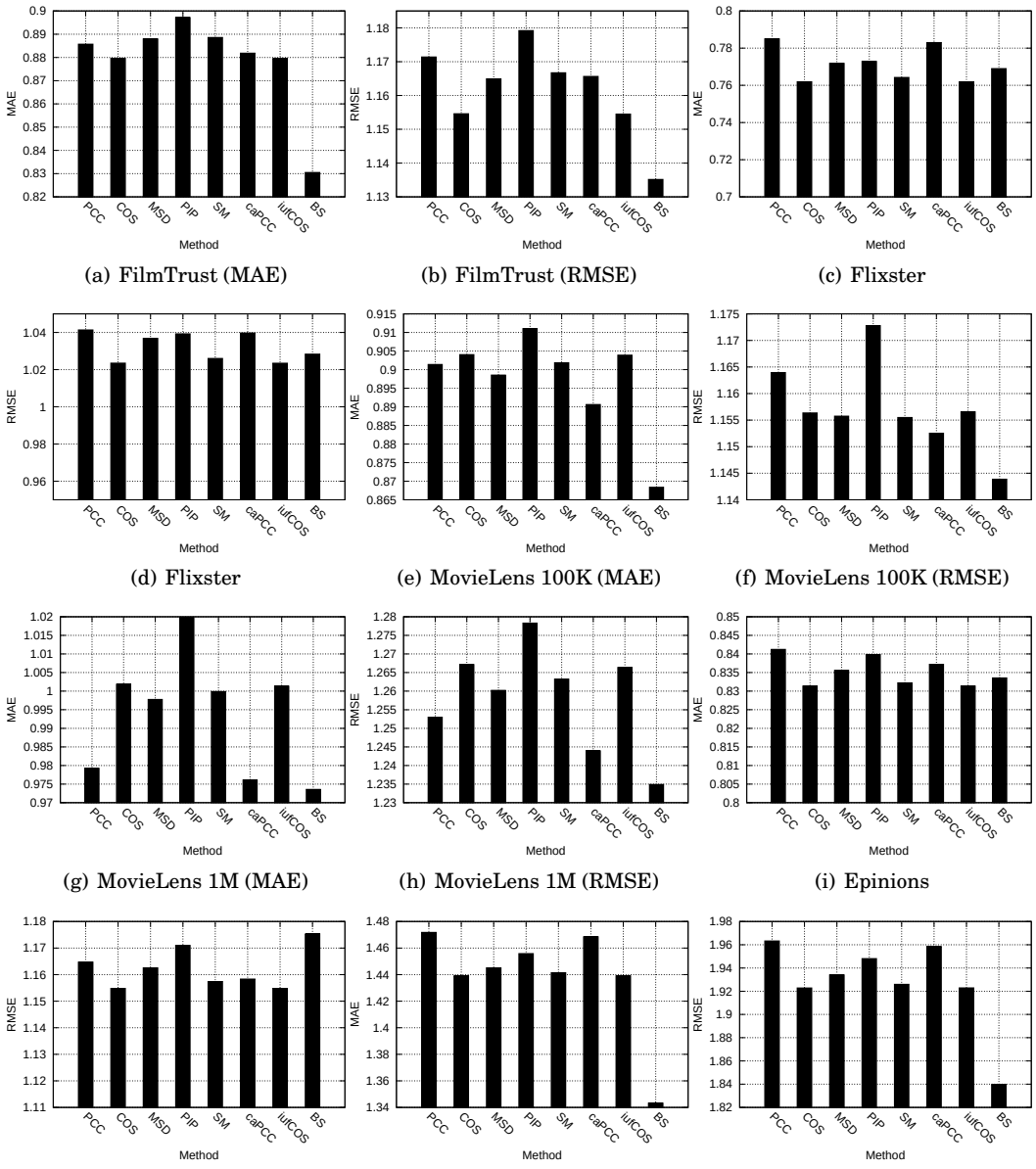
(g) BookCrossing (MAE) (h) BookCrossing (RMSE)

Fig. 7. The predictive accuracy of comparative approaches on cold users

in certain cases where our approach does not significantly outperform all the other methods, the performance by BS is often equivalent to or only slightly worse than the *best* of the other methods. Although not specialized for cold users or niche items, our approach demonstrates its generality and good performance across all the testing views. Further studying the performance on other samples of users or items would be an interesting part of future work. Note that the generality of our approach is based on the consistent performance gains obgtained across the six data sets we used in the experiments. It is possible that our approach may not outperform the other measures on some other data sets we have not tested yet.

Nevertheless, it is worth noting that the user-based K-NN method used for our comparison of similarity measures is generally not competitive with advanced model-based approaches, such as matrix factorization models[13], in terms of predictive accu-

[13]However, we also notice that on FilmTrust our approach, i.e., Bayesian similarity-based KNN is able to achieve competitive performance ($K = 30$, see Figure 6(a)) with BiasedMF [Koren 2010] the performance of which is reported at http://librec.net/example.html on the same data set.



(a) FilmTrust (MAE) (b) FilmTrust (RMSE) (c) Flixster

(d) Flixster (e) MovieLens 100K (MAE) (f) MovieLens 100K (RMSE)

(g) MovieLens 1M (MAE) (h) MovieLens 1M (RMSE) (i) Epinions

(j) Epinions (k) BookCrossing (MAE) (l) BookCrossing (RMSE)

Fig. 8. The predictive accuracy of comparative approaches on niche items

racy[14] [Koren 2010]. However, the user-based K-NN method is suitable for the present work because our main objective is to investigate the effectiveness of the Bayesian similarity measure (in comparison with others), rather than to justify that a user-based K-NN with Bayesian similarity can surpass all the other recommendation methods.

[14]A detailed comparison between UserKNN and matrix factorization methods is reported at: librec.net/example.html

We would like to stress again that similarity measures can also be used in model-based approaches such as those of Ma et al. [2011] and Shi et al. [2013]. Furthermore, the research line of recommendation models for implicit feedback [Pan et al. 2015b,a] is also beyond the discussion of this article.

The main idea of our Bayesian similarity is to take into account the importance of a number of evidence weighting factors in measuring user similarity. In principle, the same basic idea can be applied to measure item similarity by reformulating the weighting factors from the perspective of items rather than users. For example, the rating consistency can be modelled based on reliable users rather than reliable items, and the system bias is also applicable to item similarity. Designing a proper item similarity measure is an interesting topic and itself can be a separate line of research.

6. CONCLUSION AND FUTURE WORK

This article 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. We stressed the importance of evidence weights for the similarity computation and introduced three different evidence factors. We showed that an effective combination of these factors can achieve the best predictive accuracy. In addition, we found that removing chance correlation can significantly improve the computed user similarity, and that only a very limited or ignorable system bias may be caused by our method. Using typical examples, we exemplified that our Bayesian measure was capable of addressing the four issues of traditional similarity measures (i.e., the Pearson correlation coefficient and cosine similarity). More generally, we empirically analyzed the trends of these measures, and found that our method can generate more realistic and distinguishable similarity measurements. Finally, the experimental results based on six real-world data sets further demonstrated the robust effectiveness of our method in comparison with traditional and contemporary measures in terms of predictive accuracy.

The present work stresses the importance of factors to investigate evidence weights in order to better model user similarity. However, a number of parameters need to be configured according to the characteristics of data sets, e.g., c in Equation 4. We would like to further study how to better determine the values of these parameters.

Our approach only relies on numerical ratings to model user correlation and hence it can be applied into many other domains, such as biochemistry [Luo et al. 2015], image processing [Huang et al. 2015], information retrieval [Wang et al. 2015] and social media [Anderson et al. 2012]. 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]. In addition, it would be also beneficial to study the effects of different similarity measures on the other samples of users and items. Lastly, we would like to further validate our approach in the case of other recommendation tasks, such as top-N item recommendation.

Acknowledgement

This work is supported by the MoE AcRF Tier 2 Grant M4020110.020, and the Institute for Media Innovation at Nanyang Technological University, Singapore. We gratefully thank the reviewers for their constructive comments, and also thank the reviewers of the preliminary version that appeared at IJCAI'13.

REFERENCES

G. Adomavicius and J. Zhang. 2012. Impact of data characteristics on recommender systems performance. *ACM Transactions on Management Information Systems*

(TMIS) 3, 1 (2012), 3.
H.J. Ahn. 2008. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences* 178, 1 (2008), 37–51.
A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. 2012. Effects of user similarity in social media. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM)*. 703–712.
J. Bobadilla, F. Ortega, and A. Hernando. 2012. A collaborative filtering similarity measure based on singularities. *Information Processing & Management* 48, 2 (2012), 204–217.
J.S. Breese, D. Heckerman, C. Kadie, and others. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI)*. 43–52.
F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso. 2011. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web (TWEB)* 5, 1 (2011), 2.
L. Candillier, F. Meyer, and F. Fessant. 2008. Designing specific weighted similarity measures to improve collaborative filtering systems. In *Proceedings of the 8th Industrial Conference on Advances in Data Mining (ICDM)*. 242–255.
C. Desrosiers and G. Karypis. 2011. A comprehensive survey of neighborhood-based recommendation methods. *Recommender Systems Handbook* (2011), 107–144.
G. Guo, J. Zhang, and D. Thalmann. 2012. 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)*. 114–125.
G. Guo, J. Zhang, and N. Yorke-Smith. 2013. A Novel Bayesian Similarity Measure for Recommender Systems. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*. 2619–2625.
Y. Huang, X. Chen, J. Zhang, D. Zeng, D. Zhang, and X. Ding. 2015. Single-trial {ERPs} denoising via collaborative filtering on {ERPs} images. *Neurocomputing* 149, Part B (2015), 914 – 923.
Y. Koren. 2010. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 4, 1 (2010), 1:1–1:24.
N. Lathia, S. Hailes, and L. Capra. 2007. Private distributed collaborative filtering using estimated concordance measures. In *Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys)*. 1–8.
N. Lathia, S. Hailes, and L. Capra. 2008. The effect of correlation coefficients on communities of recommenders. In *Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC)*. 2000–2005.
B. Li, X. Zhu, R. Li, C. Zhang, X. Xue, and X. Wu. 2011. Cross-domain collaborative filtering over time. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*. 2293–2298.
Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning (ICML)*, Vol. 98. 296–304.
X. Luo, Z. Ming, Z. You, S. Li, Y. Xia, and H. Leung. 2015. Improving network topology-based protein interactome mapping via collaborative filtering. *Knowledge-Based Systems (KBS)* 90 (2015), 23–32.
H. Ma, I. King, and M.R. Lyu. 2007. 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)*. 39–46.
H. Ma, D. Zhou, C. Liu, M.R. Lyu, and I. King. 2011. Recommender systems with social regularization. In *Proceedings of the 4th ACM International Conference on*

Web Search and Data Mining (WSDM). 287–296.
B.K. Mohan, B.J. Keller, and N. Ramakrishnan. 2007. Scouts, promoters, and connectors: The roles of ratings in nearest neighbor collaborative filtering. *ACM Transactions on the Web (TWEB)* 1, 2 (2007), 8.
A. O'Hagan. 2004. Bayesian statistics: principles and benefits. *Frontis* 3 (2004), 31–45.
F. Ortega, J.L. Sánchez, J. Bobadilla, and A. Gutiérrez. 2013. Improving collaborative filtering-based recommender systems results using Pareto dominance. *Information Sciences* 239, 0 (2013), 50 – 61.
W. Pan, Z. Liu, Z. Ming, H. Zhong, X. Wang, and C. Xu. 2015a. Compressed Knowledge Transfer via Factorization Machine for Heterogeneous Collaborative Recommendation. *Knowledge-Based Systems (KBS)* 85 (2015), 234 – 244.
W. Pan, H. Zhong, C. Xu, and Z. Ming. 2015b. Adaptive Bayesian personalized ranking for heterogeneous implicit feedbacks. *Knowledge-Based Systems (KBS)* 73 (2015), 173–180.
Y. Ren, G. Li, J. Zhang, and W. Zhou. 2012. The efficient imputation method for neighborhood-based collaborative filtering. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*. 684–693.
S.J. Russell and P. Norvig. 2009. *Artificial Intelligence: A Modern Approach* (third ed.). Prentice Hall Englewood Cliffs, NJ.
A. Said, B.J. Jain, and S. Albayrak. 2012. Analyzing weighting schemes in collaborative filtering: Cold start, post cold start and power users. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC)*. 2035–2040.
U. Shardanand and P. Maes. 1995. Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (SIGCHI)*. 210–217.
Y. Shi, M. Larson, and A. Hanjalic. 2009. 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)*. 125–132.
Y. Shi, M. Larson, and A. Hanjalic. 2013. Mining contextual movie similarity with matrix factorization for context-aware recommendation. *ACM Transactions on Intelligent Systems and Technology (TIST)* 4, 1 (2013), 16.
J. Wang, J.Z. Huang, J. Guo, and Y. Lan. 2015. Recommending high-utility search engine queries via a query-recommending model. *Neurocomputing* (2015).