Social Recommendation with Strong and Weak Ties

Xin Wang[†‡] Wei Lu[§] Martin Ester[‡] Can Wang[†] Chun Chen[†]

[†]Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, China
[‡]School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada
[§]Department of Computer Science, University of British Columbia, Vancouver, B.C., Canada
[‡]{xwa49,ester}@cs.sfu.ca [§]welu@cs.ubc.ca [†]{xinwang,wcan,chenc}@zju.edu.cn

ABSTRACT

With the explosive growth of online social networks, it is now well understood that social information is highly helpful to recommender systems. Social recommendation methods are capable of battling the critical cold-start issue, and thus can greatly improve prediction accuracy. The main intuition is that through trust and influence, users are more likely to develop affinity toward items consumed by their social ties. Despite considerable work in social recommendation, little attention has been paid to the important distinctions between strong and weak ties, two well-documented notions in social sciences. In this work, we study the effects of distinguishing strong and weak ties in social recommendation. We use neighbourhood overlap to approximate tie strength and extend the popular Bayesian Personalized Ranking (BPR) model to incorporate the distinction of strong and weak ties. We present an EM-based algorithm that simultaneously classifies strong and weak ties in a social network w.r.t. optimal recommendation accuracy and learns latent feature vectors for all users and all items. We conduct extensive empirical evaluation on four real-world datasets and demonstrate that our proposed method significantly outperforms state-of-the-art pairwise ranking methods in a variety of accuracy metrics.

1. INTRODUCTION

Recommender systems are ubiquitous in our digital life. They play a significant role in numerous Internet services and applications such as electronic commerce (Amazon and eBay), on-demand video streaming (Netflix and Hulu), as well as social networking ("People You May Know" feature of LinkedIn and Facebook). A key task is to model user preferences and to suggest, for each user, a personalized list of items that the user has not experienced, but are deemed highly relevant to her.

Lots of recommendation techniques have been proposed in the literature [15, 43]. When explicit feedback (numerical ratings) is available, model-based collaborative filtering is among the most effective methods, e.g., low-rank matrix factorization [29]. We refer the reader to Section 2 or references therein for more details.

However, when explicit feedback is not readily unavailable, we may only have access to implicit feedback [24] derived from user actions such as viewing videos, clicking links, listening to songs, etc. In fact, implicit feedback is more abundant than explicit [24,42] in practice. Although collaborative filtering approaches can be adapted [24], pairwise ranking methods have gained more traction lately [30, 39, 41, 42, 48]. This approach focuses on learning the order of items (in user preferences). The Bayesian Personalized Ranking (BPR) framework [42] is a fundamental pairwise ranking method. In a nutshell, the core idea is to learn a personalized ranking for each user based on the assumption that a user prefers an observed item over all non-observed items. Here, an observed item refers to any item that has been consumed by the user. When the context is clear, we will use "observed" or "consumed" items interchangeably. In [42], the authors further show that many scoring methods can be integrated into BPR to learn the rankings, including matrix factorization.

A critical yet common issue faced by recommender systems is *data sparsity*, because the number of items is typically huge (e.g., hundreds of thousands) but users normally only consume a very small subset of items. An even more challenging problem related to data sparsity is that when new users join in a system, they have no history records which can be utilized by the recommender systems to learn their preferences. This leads to the cold-start problem and may result in suboptimal recommendations. To mitigate this issue, many methods have been proposed to leverage social network information in recommender systems [25, 26, 32–35, 46–48], bringing about the field of *social recommendation*. Specifically for BPR, Zhao et al. [48] propose the Social BPR (SBPR) model which further assumes that among all non-observed items, a user prefers those consumed by their social connections (or ties) to the rest. We refer the reader to Section 2 for more details.

Although there exists previous work that aims at predicting tie strength with social media [20] and analyzing roles of tie strength in Q&A online networks [40], to the best of our knowledge, there has been no systematic study on social tie strength and types in the context of recommender systems, and more importantly, the extent to which different social ties affect the quality of recommendations. In his influential paper [21], Granovetter introduces different types of social ties (strong, weak, and absent), and concludes that weak ties are actually the most important reason for new information or innovations to spread over social networks. In [22], through surveys and interviews, Granovetter reports that many job seekers find out useful information about new jobs through personal contacts. Perhaps surprisingly, many of those personal contacts are acquaintances (weak ties) as opposed to close friends (strong ties) [18, 22].

These insights from social sciences motivate us to study whether distinguishing between strong and weak ties would make a difference for social recommendations in terms of prediction accuracy (e.g., metrics such as precision, recall, and AUC). However, two

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983701

major challenges arise. First, how to learn the label of each tie (strong or weak) in a given social network? The sociology literature [21, 22] typically assumes the *dyadic hypothesis*: the strength of a tie is determined solely by the interpersonal relationship between two individuals, irrespective of the rest of the network. For instance, Granovetter uses the frequency of interactions to classify strong and weak ties [22]. This is simple and intuitive, but it requires user activity data that is hardly available to the public in modern online social networks for security and privacy reasons[1]. Second, assuming a reliable classification algorithm for learning strong and weak ties, how can we effectively incorporate such knowledge into existing ranking methods to improve recommendation accuracy?

In this work, we tackle both challenges head on. We first adopt Jaccard's coefficient, a feature intrinsic to the network topology, to compute tie strength [31, 37]. Intuitively, Jaccard captures the extent to which those users' friendship circles overlap. Our choice is endorsed by the studies on a large-scale mobile call graph by Onnela et al. [37] (more details in Section 3). We define ties as strong if their Jaccard's coefficient is above some threshold, and as weak otherwise. Note that the optimal threshold w.r.t. recommendation accuracy will be learnt from the data.

Next, we extend the BPR model and propose a unified learning framework that simultaneously (i) classifies strong and weak ties w.r.t. optimal recommendation accuracy and (ii) learns a ranking model that effectively leverages the learned tie types. We employ the Expectation-Maximization algorithm [17] to alternatively learn types of social ties and other model parameters including the latent feature vector for each user and each item. Our experiments on four real-world datasets clearly demonstrate the superiority of our method over state-of-the-art methods.

To summarize, we make the following contributions.

- We recognize the effects of strong and weak social ties that are evident in the sociology literature, and propose to incorporate these notions into social recommendation (Section 3).
- We propose a more fine-grained categorization of user-item feedback for Bayesian Personalized Ranking (BPR) by leveraging the knowledge of tie strength and tie types (Section 4).
- We present an EM-style algorithm to simultaneously learn the optimal threshold w.r.t. recommendation accuracy for classifying strong and weak ties, as well as other parameters (Section 5) in our extended BPR model.
- We carry out extensive experiments on four real-world datasets and show that our solution significantly outperforms existing methods in various accuracy metrics such as precision and recall (Section 6).

To the best of our knowledge, this is the first work recognizing the important distinctions between strong and weak ties and leveraging them to improve social recommendation.

Before proceeding further, we now formalize the problem studied in this paper. Consider a recommender system, and let $\mathcal{U}$ and $\mathcal{I}$ denote the set of users and items, respectively. There is also a social network connecting the users, represented by an undirected graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, where each node $u \in \mathcal{U}$ represents an individual user and each edge $(u, v) \in \mathcal{E}$ indicates a tie between users u and v. We know the set of items consumed by each user u, and our task is to produce a personalized ranking (a total ordering of all items), denoted $\succcurlyeq_u$, for all $u \in \mathcal{U}$.

[1]https://en.wikipedia.org/wiki/Privacy_concerns_with_social_networking_services

2. RELATED WORK

Substantial work has been done in recommender systems during the past two decades. For systems with explicit feedback (numerical ratings), *Collaborative Filtering (CF)* has become the standard approach for its accuracy and scalability. In general, there are two kinds of CF methods: memory-based and model-based. Representative memory-based methods contain k-Nearest Neighbour (kNN) user-user cosine similarity and item-item cosine similarity [15]. Typical model-based methods include low-rank matrix factorization [28, 29, 44] which is also widely-adopted for other tasks such as feature selection [2] and can be combined with various techniques like deep learning for recommendation as well [1]. Since this work focuses on implicit feedback, we simply refer the reader to excellent surveys and monographs for more details [15, 43].

Recommender Systems with Implicit Feedback. Considerable work has been done to address the problem of how to use just implicit feedback to generate high quality recommendations. Oard and Kim [36] identified several data sources to gather implicit feedback and suggested two types of recommendation strategies. The first strategy is to infer explicit ratings that users are likely to produce and adopt available methods for explicit feedback. The second one is directly infer user preferences without converting implicit feedback to ratings. Das et al. [16] presented an online recommendation algorithm for Google News where only click history of each user is available (hence implicit). They describe a linear model that combines three recommendation algorithms: collaborative filtering using MinHash clustering, probabilistic Latent Semantic Indexing (pLSI), and co-visitation counts. Hu et al. [24] proposed to transform implicit feedback into two paired quantities: preferences and confidence levels and use both of them to learn a latent factor model. Unlike matrix factorization for explicit feedback, their model takes all user-item pairs, including non-observed items, as an input and is later extended for other recommendation tasks by others [3]. Scalable learning algorithms are proposed to address the (potentially) huge amount of input. Pan et al. [38] uses weighted low-rank approximation and sampling techniques. First, different weights are assigned to the error terms of observed items and non-observed items in the objective function. Second, they sample non-observed items as negative feedback, instead of using all of them.

All the aforementioned methods are often referred to as *point-wise*, since they learn absolute preferences and then produce top-K recommendations by simply sorting items by their scores in descending order. Rendle et al. [42] proposed a novel *pairwise* learning method called Bayesian Personalized Ranking (BPR). Here the focus is shifted to the learning of *relative* preferences. BPR trains on pairs of items and the objective is to maximize the posterior likelihood of optimal personalized ranking, in which the assumption is that for each user, observed items are preferred over non-observed ones. Empirical results in [42] demonstrate that BPR coupled with matrix factorization or kNN indeed outperform point-wise methods proposed in [24, 38]. Recently, Rendle and Freudenthaler [41] introduced a more sophisticated sampling technique to improve the convergence rate of BPR learning.

Social Ties in Social Media. Social ties have been widely studied in social science [10, 11, 21, 22]. There has been recent work in Computer Science that pays attention to tie strength in social media [4–8, 20, 40, 45] and with demographic data [9]. Specifically, Gilbert et al. [20] proposed a method for predicting social tie strength and conducted a user-study experiment with over 2000 social media ties. Wu et al. [45] performed a regression analysis to detect the professional and personal closeness between employees in an IBM enterprise social network. Panovich et al. [40] employed

Wu's method to examine roles of tie strength in question and answer online networks.

However, none of the above work leverages the theory of social ties into recommender systems. On the other hand, existing work in social recommendations (discussed below) do not take different types of social ties into consideration.

Social Recommendation. In a nutshell, social recommendation aims to exploit the effects of trust and influence to address the cold-start problem, which may cause traditional CF methods to fail due to lack of feedback data from cold-start users. Jamali et al. [25] reported that in the Epinions dataset, about 50% of the users are deemed cold-start (who rated less than five items). Considerable work has been done in this domain [25, 26, 32–35, 46–48]. However, the overwhelming majority of those social recommendation methods are designed for explicit feedback systems, with few exceptions.

Recently, Zhao et al. [48] extended the BPR framework by further assuming that amongst all non-observed items, a user would prefer items consumed by her social ties over the rest (which we call "social items" hereafter for simplicity). In their SBPR model, for each user u, the relative preference between any self-consumed item i and any social item j is discounted by the number of u's ties who consumed j. That is, the more ties consumed j, the smaller the gap is between i and j in the eyes of u. They also discussed an alternative, opposite assumption, i.e., the social items are perceived even more negatively than "non-social" items. Their experiments showed that this alternative SBPR model is not as good as the first one. We depart from SBPR by making orthogonal social-aware extensions to BPR. In particular, we recognize the importance of distinguishing between strong and weak ties and extend the BPR model by incorporating such distinctions. The key difference lies in the ranking of social items. In SBPR, social items are ranked based on the number of friends who consumed the item, while in our model, the ranking is based on tie types. Our empirical results demonstrate that our new model significantly outperforms SBPR and the vanilla BPR in terms of prediction accuracy, as measured by six different metrics including precision, recall, etc (Section 6).

3. STRONG AND WEAK TIES

The theory of strong and weak ties has first been formulated by Granovetter [21]. In terms of interpersonal relationship, strong ties correspond to close friends that have high frequency of interactions, while weak ties correspond to acquaintances. In terms of network structures, strong ties tend to be clustered in a dense subgraph (e.g., the triangles (u, v, w) and (x, y, z) in Figure 1), while weak ties tend to be "bridges" connecting two different connected components, e.g., (u, x) in Figure 1.

There is an elegant connection between the above two perspectives [18,21]. First, we say that a node u satisfies the *Strong Triadic Closure property* if it does not violate the following condition: u has two strong ties v and w but there exists no edge between v and w. Furthermore, if a node u satisfies this property and is involved in at least two strong ties, then any local bridges[2] in which it is involved must be a weak tie.

It is well understood that since weak ties typically do not belong to the same social circle, they have access to different information sources, and thus the information exchange have more novelty [18, 21, 22]. Applying this insight to the context of social recommendation, our intuition is that the items previously consumed by weak-tie friends *might* be of more interest to the user. For example, a researcher may not be able to discover many interesting new papers from her close collaborators, as they tend to focus on the same topic and read the same set of papers. Instead, she may find papers cited by other less frequent collaborators more appealing.

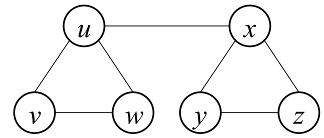


Figure 1: A sample social network

To incorporate the distinction between strong and weak ties into social recommendation, we first need to be able to define and compute tie strength, and then classify ties. Several possibilities exist. First, as mentioned in Section 1, sociologists use dyadic measures such as frequency of interactions [22]. However, this method is not generally applicable due to lack of necessary data.

An alternative approach relies on community detection. Specifically, it first runs a community detection algorithm to partition the network $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ into several subgraphs. Then, for each edge $(u, v) \in \mathcal{E}$, if u and v belong to the same subgraph, then it is classified as a strong tie; otherwise a weak tie. However, a key issue is that although numerous community detection algorithms exist [19], there is no consensual gold standard so it is unclear which one to use. Furthermore, if a "bad" partitioning (w.r.t. prediction accuracy) is produced and given to the recommender system as input, it would be very difficult for the recommender system to recover. In other words, the quality of recommendation would depend on an *exogenous* community detection algorithm that the recommender system *has no control over*. Hence, this approach is undesirable.

In light of the above, we resort to node-similarly metrics that measure neighbourhood overlap of two nodes in the network. The study of Onnela et al. [37] provides empirical confirmation of this intuition: they find that (i) tie strength is in part determined by the local network structure and (ii) the stronger the tie between two users, the more their friends overlap. In addition, unlike frequency of interactions, node-similarity metrics are intrinsic to the network, requiring no additional data to compute. Also, unlike the community detection based approach, we still get to choose a tie classification method that best serves the interest of the recommender system.

More specifically, we use Jaccard's coefficient, a simple measure that effectively captures neighbourhood overlap. Let $\mathsf{strength}(u, v)$ denote the tie strength for any $(u, v) \in \mathcal{E}$. We have:

$$\mathsf{strength}(u, v) =_{\text{def}} \frac{|\mathcal{N}_u \cap \mathcal{N}_v|}{|\mathcal{N}_u \cup \mathcal{N}_v|} \quad \textit{(Jaccard)}, \tag{1}$$

where $\mathcal{N}_u \subseteq \mathcal{U}$ (resp. $\mathcal{N}_v \subseteq \mathcal{U}$) denotes the set of ties of u (resp. v). If $\mathcal{N}_u = \mathcal{N}_v = \emptyset$ (i.e., both u and v are singleton nodes), then simply define $\mathsf{strength}(u, v) = 0$. By definition, all strengths as defined in Equation (1) fall into the interval $[0, 1]$. This definition has natural probabilistic interpretations: Given two arbitrary users u and v, their Jaccard's coefficient is equal to the probability that a randomly chosen tie of u (resp. v) is also a tie of v (resp. u) [31].

Thresholding. To distinguish between strong and weak ties, we adopt a simple *thresholding* method. For a given social network graph $\mathcal{G}$, let $\theta_{\mathcal{G}} \in [0, 1)$ denote the threshold of tie strength such that

$$(u, v) \text{ is } \begin{cases} \text{strong}, & \text{if } \mathsf{strength}(u, v) > \theta_{\mathcal{G}}; \\ \text{weak}, & \text{if } \mathsf{strength}(u, v) \le \theta_{\mathcal{G}}. \end{cases} \tag{2}$$

Let $\mathcal{W}_u =_{\text{def}} \{v \in \mathcal{U} : (u, v) \in \mathcal{E} \wedge \mathsf{strength}(u, v) \le \theta_{\mathcal{G}}\}$

[2] (u, v) is a local bridge if the deletion of this edge results in u and v to have a shortest path distance of 3 or longer.

denote the set of all weak ties of u. Similarly, $\mathcal{S}_u =_{\text{def}} \{v \in \mathcal{U} : (u, v) \in \mathcal{E} \wedge \mathsf{strength}(u, v) > \theta_{\mathcal{G}}\}$ denotes the set of all strong ties of u. Clearly, $\mathcal{W}_u \cap \mathcal{S}_u = \emptyset$ and $\mathcal{W}_u \cup \mathcal{S}_u = \mathcal{N}_u$.

In our framework, the value of $\theta_{\mathcal{G}}$ is *not* hardwired, but rather is left for our model to learn (Section 5), such that the resulting classification of strong and weak ties in $\mathcal{G}$, together with other learned parameters of the model, leads to the best accuracy of recommendations.

Finally, we remark that other node-similarity metrics can also be used to define tie strength, e.g., Adamic-Adar [14] and Katz score [27]. However, we note that the exact choice amongst these node-similarly metrics is not the primary focus of this paper and is orthogonal to our proposed learning framework.

4. THE TBPR MODEL: BPR WITH STRONG AND WEAK TIES

In this section, we present our TBPR (*BPR with Strong and Weak Ties*) model which incorporates the distinction of strong and weak ties into BPR and ranks social items based on types of ties.

4.1 Categorizing Items

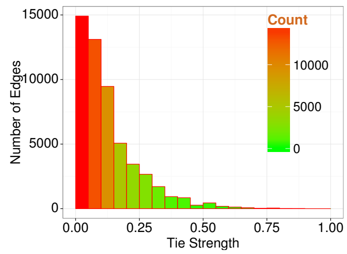
Having defined strong and weak ties, we are now ready to present a key element in our TBPR model: For every user we categorize all items into five types using the knowledge of strong and weak ties, which we then exploit in our TBPR model. Here, we provide a fine-grained categorization of non-observed items, especially the social items, by leveraging strong and weak tie information derived from the social network graph $\mathcal{G}$. The proposed categorization is as follows.

1. **Consumed Items.** For all $u \in \mathcal{U}$, let $\mathcal{C}_u^{self} \subseteq \mathcal{I}$ denote the set of items consumed by u itself.
2. **Joint-Tie-Consumed (JTC) Items.** Any item $i \in \mathcal{I} \setminus \mathcal{C}_u^{self}$ that has been consumed by at least one strong tie of u and one weak tie of u belongs to this category. We denote this set by $\mathcal{C}_u^{joint} = \{i \in \mathcal{I} \setminus \mathcal{C}_u^{self} : \exists v \in \mathcal{S}_u \text{ s.t. } i \in \mathcal{C}_v^{self} \wedge \exists w \in \mathcal{W}_u \text{ s.t. } i \in \mathcal{C}_w^{self}\}$
3. **Strong-Tie-Consumed (STC) Items.** If an item $i \in \mathcal{I} \setminus \mathcal{C}_u^{self}$ is consumed by at least one strong tie of u, but not by u itself or weak ties, then it belongs to this category. We denote this set by $\mathcal{C}_u^{strong} = \{i \in \mathcal{I} \setminus \mathcal{C}_u^{self} : \exists v \in \mathcal{S}_u \text{ s.t. } i \in \mathcal{C}_v^{self} \wedge \nexists w \in \mathcal{W}_u \text{ s.t. } i \in \mathcal{C}_w^{self}\}$.
4. **Weak-Tie-Consumed (WTC) Items.** This category can be similarly defined: $\mathcal{C}_u^{weak} = \{i \in \mathcal{I} \setminus \mathcal{C}_u^{self} : \nexists v \in \mathcal{S}_u \text{ s.t. } i \in \mathcal{C}_v^{self} \wedge \exists w \in \mathcal{W}_u \text{ s.t. } i \in \mathcal{C}_w^{self}\}$.
5. **Non-Consumed Items.** This category contains the rest of the items (not consumed by u or any of u's ties): $\mathcal{C}_u^{none} = \{(u, i) : \nexists x \in \mathcal{S}_u \cup \mathcal{W}_u \text{ s.t. } i \in \mathcal{C}_x^{self}\}$.

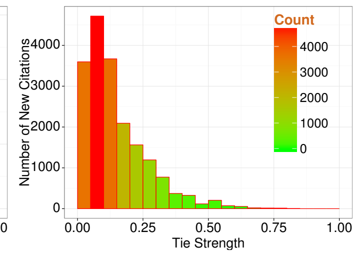
Clearly, for all $u \in \mathcal{U}$, $\mathcal{C}_u^{self} \cup \mathcal{C}_u^{joint} \cup \mathcal{C}_u^{strong} \cup \mathcal{C}_u^{weak} \cup \mathcal{C}_u^{none} = \mathcal{I}$. In addition, those five sets are pairwise disjoint. Note that the union of JTC, STC, and WTC items is the set of all social items for user u.

4.2 Ordering Item Types

We now describe our TBPR model which distinguishes between the aforementioned five types of items for every user. Same as the original BPR, we assume no particular item scoring method [42]. However, for ease of exposition and its effectiveness, we use low-rank matrix factorization [29], which is considered as a state-of-the-art collaborative filtering method in the literature.



(a) Tie strength



(b) New citations

Figure 2: Histograms of tie strength and new citations for the DBLP dataset

Assume that every user and every item in the system are represented by a d-dimensional latent feature vector: let $\mathbf{P}_u \in \mathbb{R}^d$ and $\mathbf{Q}_i \in \mathbb{R}^d$ denote the feature vector for an arbitrary user u and an arbitrary item i, respectively. Here d is the number of latent features. The inner product between a user feature vector and an item feature vector measures the estimated affinity this user has toward the item (a.k.a. predicted personalized score), denoted by $\hat{r}_{ui} =_{\text{def}} \langle \mathbf{P}_u, \mathbf{Q}_i \rangle$. Since we deal with binary feedback in this work, we have $\hat{r}_{ui} \in [0, 1]$ for all $u \in \mathcal{U}$ and all $i \in \mathcal{I}$.

In this paper, the proposed TBPR model imposes a total ordering of the five item types that specifies user preference. Indicated by the good performance of BPR and its variants [39, 48], we also assume that users prefer consumed items over others. Hence, consumed items rank at the top of the ordering. Next, it is an open question that whether users prefer WTC items to STC items, or vice versa. Although we mentioned in Section 1 that the sociology literature has suggested that weak ties are responsible for more novel information to spread over the social network, it does not automatically mean that WTC items are preferred.

To investigate the above question, we conduct a case study using co-authorship and citation data extracted from the DBLP Computer Science Bibliography (http://dblp.uni-trier.de/db/). The DBLP dataset together with four other public datasets (Epinions, Douban and Ciao) will be used to evaluate the performances of different methods later in the experiment section. Recently some work has been done on recommending papers to read or cite using the DBLP dataset [13], making it another appropriate experimental dataset for us to test the performances of different recommendation algorithms. Furthermore, the DBLP website provides researchers with an API so that they can crawl their own datum from the database for the purpose of scientific research, which enables the possibility for us to obtain the information about the evolution of co-authorship network and conduct this case study based on the assumption that co-authorship network and citations follow a similar pattern to social network and other user-item consumption behaviours.

The network graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ is constructed as follows. First, each node $v \in \mathcal{U}$ corresponds to an author satisfying both (i) she co-authored at least ten papers and (ii) at least one of her papers was published in or after 2009. If two authors u and v have co-authored at least one paper before 2009, then there is an undirected edge $(u, v) \in \mathcal{E}$. As a result, the graph contains 13.6K nodes and 107K edges.

Figure 2(a) shows the distribution of tie strength as computed by Equation (1). By definition, if two authors u and v have a strong tie, then a relatively large overlap exists amongst their collaborators. As we can see, the distribution is skewed toward weak tie strength.

Next, we analyze the citation data to see whether researchers are more likely to cite papers that were previously cited by their weak ties as opposed to strong ties, or vice versa. We are interested in the case of *follow-up citations*. For example, for any $(u, v) \in \mathcal{E}$, if there exists a paper cited by v but not by u before 2009, and u cited this paper in or after 2009, then we say that u made one follow-up citation to v. Note that, this definition eliminates all citations occurred in papers co-authored by u and v, which are not interesting to us. Figure 2(b) plots the number of follow-up citations against tie strength. We can see that this distribution is also heavily skewed toward weak tie strength. This suggests that from the perspective of absolute number, researchers indeed tend to cite papers that are previously cited by their weak ties.

Our two plots above are very consistent with those plots (with Y axis showing the probability of job help instead of the number of citations) presented in [12], a recent work by Gee et al. on how strong ties and weak ties relate to job finding on Facebook's social network. Gee et al. use both mutual interactions and node similarity (similar to Jaccard's coefficient) to measure tie strength and find results to be similar for both kinds of measures, which provides further support for using Jaccard's coefficient as our tie strength measure. Readers may refer to [12] for more details. A conclusion in their work is that weak ties are important collectively because of their quantity, and strong ties are important individually because of their quality. Reflected in our TBPR model, we can say that the sets of WTC items are more helpful than the sets of STC items and an individual STC item may be more helpful than an individual WTC item. Thus giving the WTC items a higher probability to be exposed (recommended) to users (i.e., ranking WTC items ahead of STC items) should help to discover potentially more interesting items. On the other hand, we also explore the opposite case of users ranking STC items ahead of WTC items. As such, we test both ranking strategies for completeness, in what follows, we present two variants of our TBPR model.

4.3 Two Variants of TBPR

We are now ready to define two variants of TBPR, which differ in the preference between WTC and STC items.

TBPR-W (Preferring Weak Ties). Mathematically, under the hypothesis that WTC items are preferred to STC items, the complete ordering is thus:

$$i \succcurlyeq_u j, \text{ if } \begin{cases} i \in \mathcal{C}_u^{self} \wedge j \in \mathcal{C}_u^{joint} & \text{or} \\ i \in \mathcal{C}_u^{joint} \wedge j \in \mathcal{C}_u^{weak} & \text{or} \\ i \in \mathcal{C}_u^{weak} \wedge j \in \mathcal{C}_u^{strong} & \text{or} \\ i \in \mathcal{C}_u^{strong} \wedge j \in \mathcal{C}_u^{none}. \end{cases} \tag{3}$$

Note that Equation (3) gives a total ordering of the five types due to transitivity, e.g., it also holds that $i \succcurlyeq_u j$ if $i \in \mathcal{C}_u^{self}$ and $j \in \mathcal{C}_u^{none}$.

TBPR-S (Preferring Strong Ties). Alternatively, we may also assume that users prefer STC items to WTC items, in which case the ordering can be expressed as:

$$i \succcurlyeq_u j, \text{ if } \begin{cases} i \in \mathcal{C}_u^{self} \wedge j \in \mathcal{C}_u^{joint} & \text{or} \\ i \in \mathcal{C}_u^{joint} \wedge j \in \mathcal{C}_u^{strong} & \text{or} \\ i \in \mathcal{C}_u^{strong} \wedge j \in \mathcal{C}_u^{weak} & \text{or} \\ i \in \mathcal{C}_u^{weak} \wedge j \in \mathcal{C}_u^{none}. \end{cases} \tag{4}$$

When it is clear from the context, we use the generic name TBPR to refer to both variants. The specific names TBPR-W and TBPR-S will be used when it is necessary to distinguish between them (e.g., comparisons in experimental results).

5. PARAMETER LEARNING

In this section, we present the optimization objective and an EM-style learning algorithm for our TBPR model. Without loss of generality, our presentation focuses on TBPR-W in which WTC items are preferred over STC items. The case of TBPR-S is symmetric and hence is omitted.

5.1 Optimization Objective

Let Θ denote the set of all parameters that consists of (i) the tie strength threshold $\theta_{\mathcal{G}}$ and (ii) the latent feature vectors: $\mathbf{P}_u$ for each user $u \in \mathcal{U}$ and $\mathbf{Q}_i$ for each item $i \in \mathcal{I}$. The likelihood function can thus be expressed as:

$$\begin{aligned}\mathcal{L}(\Theta) = \prod_{u \in \mathcal{U}} \Bigg(& \prod_{i \in \mathcal{C}_u^{self}} \prod_{j \in \mathcal{C}_u^{joint}} \Pr[i \succcurlyeq_u j] \\ & \prod_{j \in \mathcal{C}_u^{joint}} \prod_{w \in \mathcal{C}_u^{weak}} \Pr[j \succcurlyeq_u w] \\ & \prod_{w \in \mathcal{C}_u^{weak}} \prod_{s \in \mathcal{C}_u^{strong}} \Pr[w \succcurlyeq_u s] \\ & \prod_{s \in \mathcal{C}_u^{strong}} \prod_{k \in \mathcal{C}_u^{none}} \Pr[s \succcurlyeq_u k] \Bigg),\end{aligned} \tag{5}$$

where the probabilities are defined using the sigmoid function following common practice [42]: $\delta(x) = \frac{1}{1+\exp(-x)}$.

For instance, the probability that consumed items are preferred over JTC items can be written as follows.

$$\begin{aligned} & \Pr[i \succcurlyeq_u j] \\ & = \delta(\hat{x}_{ui} - \hat{x}_{uj}) \\ & = \frac{1}{1 + \exp(-(\hat{x}_{ui} - \hat{x}_{uj}))} \\ & = \frac{1}{1 + \exp(-\langle \mathbf{P}_u, \mathbf{Q}_i \rangle + \langle \mathbf{P}_u, \mathbf{Q}_j \rangle)}. \end{aligned} \tag{6}$$

All other probabilities except for the probability that WTC items are preferred over STC items, namely $\Pr[w \succcurlyeq_u s]$, can be defined similarly. We omit the formulas as they resemble Eq. (6) closely.

Incorporating the Tie Strength Threshold

Given a threshold $\theta_{\mathcal{G}}$, the *degree of separation* between strong ties and weak ties imposed by this threshold can be quantitatively measured using the following formula:

$$g(\theta_{\mathcal{G}}) = (\bar{t}_s - \theta_G)(\theta_G - \bar{t}_w), \tag{7}$$

where $\bar{t}_s$ is the average strength of all strong ties classified according to $\theta_{\mathcal{G}}$ and likewise $\bar{t}_w$ is the average strength of all weak ties.

A threshold $\theta_{\mathcal{G}}$ that gives a large degree of separation $g(\theta_{\mathcal{G}})$ is desirable. To incorporate the threshold into the objective function so that our TBPR model is able to learn it in a principled manner, we add a coefficient $1/g(\theta_{\mathcal{G}})$ into the probability that WTC items are preferred over STC items. More specifically, we define:

$$\begin{aligned} \Pr[w \succcurlyeq_u s] & = \delta\left(\frac{\hat{x}_{uw} - \hat{x}_{us}}{1 + 1/g(\theta_{\mathcal{G}})}\right) \\ & = \frac{1}{1 + \exp\left(-\frac{\hat{x}_{uw} - \hat{x}_{us}}{1+1/g(\theta_{\mathcal{G}})}\right)} \\ & = \frac{1}{1 + \exp\left(\frac{-\langle \mathbf{P}_u, \mathbf{Q}_w \rangle + \langle \mathbf{P}_u, \mathbf{Q}_s \rangle}{1+1/g(\theta_{\mathcal{G}})}\right)}, \end{aligned} \tag{8}$$

where we use $1 + 1/g(\theta_{\mathcal{G}})$ to discount $(\hat{x}_{uw} - \hat{x}_{us})$, the difference between u's predicted score for w and s. The intuition is that, if the current threshold $\theta_\mathcal{G}$ does not separate the strong and weak ties well enough, the likelihood that user prefers w (an WTC item given the current threshold) to s (an STC items given the current threshold) should be discounted. We use the reciprocal mainly for smoothness.

Putting It All Together

Our goal is to learn the best set of parameters that maximizes the likelihood function $\mathcal{L}(\cdot)$. This amounts to maximizing the logarithm of $\mathcal{L}(\cdot)$. Regularization terms are added to avoid overfitting:

$$r(\Theta) = \lambda_p \sum_{u\in\mathcal{U}} ||\mathbf{P}_u||_2^2 + \lambda_q \sum_{i\in\mathcal{I}} ||\mathbf{Q}_u||_2^2 + \lambda_\theta \theta_\mathcal{G}^2.$$

Putting it all together, our final maximization objective is

$$\begin{aligned}
&\mathcal{J}(\Theta) = \ln \mathcal{L}(\Theta) - r(\Theta) \\
&= \sum_{u\in\mathcal{U}} \Bigg(\sum_{i\in\mathcal{C}_u^{self}} \sum_{j\in\mathcal{C}_u^{joint}} \ln \delta(\hat{x}_{ui} - \hat{x}_{uj}) \\
&\quad + \sum_{j\in\mathcal{C}_u^{joint}} \sum_{w\in\mathcal{C}_u^{weak}} \ln \delta(\hat{x}_{uj} - \hat{x}_{uw}) \\
&\quad + \sum_{w\in\mathcal{C}_u^{weak}} \sum_{s\in\mathcal{C}_u^{strong}} \ln \delta\left(\frac{\hat{x}_{uw} - \hat{x}_{us}}{1 + 1/g(\theta_\mathcal{G})}\right) \\
&\quad + \sum_{s\in\mathcal{C}_u^{strong}} \sum_{k\in\mathcal{C}_u^{none}} \ln \delta(\hat{x}_{us} - \hat{x}_{uk}) \Bigg) \\
&\quad - \lambda_p \sum_{u\in\mathcal{U}} ||\mathbf{P}_u||_2^2 - \lambda_q \sum_{i\in\mathcal{I}} ||\mathbf{Q}_u||_2^2 - \lambda_\theta \theta_\mathcal{G}^2.
\end{aligned}$$

5.2 Learning Algorithm

We employ the *Expectation-Maximization* (EM) algorithm as well as *stochastic gradient descent* to learn the parameters Θ that maximize $\mathcal{J}(\cdot)$. In the EM algorithm, the tie strength threshold $\theta_\mathcal{G}$ is treated as a hidden parameter to be learnt from the data.

The pseudocode of the learning algorithm is presented in Algorithm 1. In the beginning, we randomly initialize the latent feature vectors for all users and all items by sampling from the uniform distribution over the interval $[0, 1]$. We initialize the tie strength threshold to be the median strength of all edges in the graph (Lines 1–3).

E-step. In each iteration t, given the current tie strength threshold $\theta_\mathcal{G}^{(t)}$, we first compute, for each user, their five categories of items (Line 6). Then, we take a total number of $100 \cdot |\mathcal{U}|$ samples as the training dataset to perform stochastic gradient descent. For each sample r, we first draw a user u uniformly at random from $\mathcal{U}$, and then draw one item from each category for this user: consumed ($\mathcal{C}_u^{self}$), JTC ($\mathcal{C}_u^{joint}$), WTC ($\mathcal{C}_u^{weak}$), STC ($\mathcal{C}_u^{strong}$), and non-consumed ($\mathcal{C}_u^{none}$) (Lines 9–14). All samples are drawn independently.

Notice that the pseudocode assumes all five item categories for all users are non-empty. If any of the sets $\mathcal{C}_u^{self}$, $\mathcal{C}_u^{joint}$, $\mathcal{C}_u^{weak}$, and $\mathcal{C}_u^{strong}$ is empty, we simply skip all relevant terms. The case of $\mathcal{C}_u^{none} = \emptyset$ is uninteresting as that would mean the user has consumed all items, and thus there is nothing left to rank for her.

Lastly, we compute the gradient of all corresponding feature vectors and perform updates (Lines 15–16). Gradients are computed using the following partial derivative formulas.

- The gradient of vector $\mathbf{P}_u$, for any user u:

$$\frac{\partial\mathcal{J}}{\partial\mathbf{P}_u} = \delta(\hat{x}_{uj} - \hat{x}_{ui})(\mathbf{Q}_i - \mathbf{Q}_j) + \delta(\hat{x}_{uw} - \hat{x}_{uj})(\mathbf{Q}_j - \mathbf{Q}_w) + \frac{\delta(\hat{x}_{us} - \hat{x}_{uw})}{1 + 1/g(\theta_\mathcal{G})}(\mathbf{Q}_w - \mathbf{Q}_s) + \delta(\hat{x}_{uk} - \hat{x}_{us})(\mathbf{Q}_s - \mathbf{Q}_k) - \lambda_p \mathbf{P}_u \tag{9}$$

Algorithm 1: Learning Algorithm for TBPR-W

Input: users $\mathcal{U}$, items $\mathcal{I}$, consumed items $\mathcal{C}_u^{self}$ for each $u \in \mathcal{U}$, social network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Output: $\Theta = \{\mathbf{P} \in \mathcal{R}^{|\mathcal{U}|\times d}, \mathbf{Q} \in \mathcal{R}^{|\mathcal{I}|\times d}, \theta_\mathcal{G}\}$

1 $\mathbf{P} \sim U(0, 1), \mathbf{Q} \sim U(0, 1)$
2 $t \leftarrow 0$ `// iteration number`
3 $\theta_\mathcal{G}^{(t)} \leftarrow$ median tie strength in $\mathcal{G}$
4 **repeat**
5   **for** $u \leftarrow 1$ **to** $|\mathcal{U}|$ **do**
6   Compute $\mathcal{C}_u^{self}, \mathcal{C}_u^{joint}, \mathcal{C}_u^{strong}, \mathcal{C}_u^{weak}, \mathcal{C}_u^{none}$ using the current tie strength threshold $\theta_\mathcal{G}^{(t)}$ `// cf. Equation (2) and categorization rules in Section 4.1`
7   **end**
8   **for** $r \leftarrow 1$ **to** $100|\mathcal{U}|$ **do**
9     $u \leftarrow$ a random user from $\mathcal{U}$
10     $i \leftarrow$ a random consumed item from $\mathcal{C}_u^{self}$
11     $j \leftarrow$ a random JTC item from $\mathcal{C}_u^{joint}$
12     $w \leftarrow$ a random WTC item from $\mathcal{C}_u^{weak}$
13     $s \leftarrow$ a random STC item from $\mathcal{C}_u^{strong}$
14     $k \leftarrow$ a random non-consumed item from $\mathcal{C}_u^{none}$
15   Compute the gradients of $\mathbf{P}_u$, $\mathbf{Q}_i$, $\mathbf{Q}_j$, $\mathbf{Q}_w$, $\mathbf{Q}_s$, and $\mathbf{Q}_k$ `// Equation (9) - Equation (14)`
16   Update the above feature vectors `// Equation (16)`
17   **end**
18   Compute $\frac{\partial\mathcal{J}}{\partial\theta_\mathcal{G}}$ `// Equation (15)`
19   $\theta_\mathcal{G}^{(t+1)} \leftarrow$ compute according to Equation (16) $t \leftarrow t + 1$
20 **until** *convergence*

- The gradient of vector $\mathbf{Q}_i$, where $i \in \mathcal{C}_u^{self}$:

$$\frac{\partial\mathcal{J}}{\partial\mathbf{Q}_i} = \delta(x_{uj} - x_{ui})\mathbf{P}_u - \lambda_q \mathbf{Q}_i \tag{10}$$

- The gradient of vector $\mathbf{Q}_j$, where $j \in \mathcal{C}_u^{joint}$:

$$\frac{\partial\mathcal{J}}{\partial\mathbf{Q}_j} = \left(\delta(x_{uw} - x_{uj}) - \delta(x_{uj} - x_{ui})\right)\mathbf{P}_u - \lambda_q \mathbf{Q}_j \tag{11}$$

- The gradient of vector $\mathbf{Q}_w$, where $w \in \mathcal{C}_u^{weak}$:

$$\frac{\partial\mathcal{J}}{\partial\mathbf{Q}_w} = \left(\frac{\delta(x_{us} - x_{uw})}{1 + 1/g(\theta_\mathcal{G})} - \delta(x_{uw} - x_{uj})\right)\mathbf{P}_u - \lambda_q \mathbf{Q}_w \tag{12}$$

- The gradient of vector $\mathbf{Q}_s$, where $s \in \mathcal{C}_u^{strong}$:

$$\frac{\partial\mathcal{J}}{\partial\mathbf{Q}_s} = \left(\delta(x_{uk} - x_{us}) - \frac{\delta(x_{us} - x_{uw})}{1 + 1/g(\theta_\mathcal{G})}\right)\mathbf{P}_u - \lambda_q \mathbf{Q}_s \tag{13}$$

- The gradient of vector $\mathbf{Q}_j$, where $j \in \mathcal{C}_u^{none}$:

$$\frac{\partial\mathcal{J}}{\partial\mathbf{Q}_k} = -\delta(x_{uk} - x_{us})\mathbf{P}_u - \lambda_q \mathbf{Q}_k \tag{14}$$

M-step. After updating the feature vectors associated with all $100|\mathcal{U}|$ samples, we update the tie strength threshold $\theta_\mathcal{G}$. The derivative can be computed as follows:

$$\frac{\partial\mathcal{J}}{\partial\theta_\mathcal{G}} = \frac{1}{100|\mathcal{U}|} \sum_{(u,w,s)} \Bigg[-\lambda_\theta\theta_\mathcal{G} + \frac{\delta(x_{us} - x_{uw})(\langle\mathbf{P}_u, \mathbf{Q}_w\rangle - \langle\mathbf{P}_u, \mathbf{Q}_s\rangle)[(\bar{t}_w + \bar{t}_s) - 2\theta_\mathcal{G}]}{[(\theta_\mathcal{G} - \bar{t}_w)(\bar{t}_s - \theta_\mathcal{G}) + 1]^2} \Bigg] \tag{15}$$

	DBLP	Ciao	Douban	Epinions
#users	13554	1141	13492	10306
#items	51877	11640	45282	109534
#non-zeros	488368	26507	2669675	375241
#ties (edges)	106730	15059	443753	230684

Table 1: Overview of datasets (#non-zeros means the number of user-item pairs that have feedback)

where (u, w, s) denotes the user, WTC item, STC item tuple sampled in one of the $100|\mathcal{U}|$ samples.

In both the E-step and M-step, the update is done using standard gradient descent:

$$x^{(t+1)} = x^{(t)} + \eta^{(t)} \cdot \frac{\partial \mathcal{J}}{\partial x}(x^{(t)}), \tag{16}$$

where $x \in \Theta$ denotes any model parameter. Finally, the algorithm terminates when the absolute difference between the losses in two consecutive iterations is less than 10^{-5}.

6. EMPIRICAL EVALUATION

In this section, we conduct extensive experiments on four real-world datasets and compare the performance of our TBPR-W and TBPR-S models with different baseline methods based on various evaluation metrics.

6.1 Experimental Settings

Datasets. We use the following four real-world datasets, whose basic statistics are summarized in Table 1.

- *DBLP.* This dataset contains information of author citation and co-author network between 1960 and 2010, which is extracted by us from the DBLP Computer Science Bibliography.
- *Ciao.* This dataset contains trust relationships between users and ratings on DVDs. It was crawled from the entire category of DVDs of a UK DVD community website http://dvd.ciao.co.uk in December, 2013, and first introduced in [23].
- *Douban.* This dataset is extracted from the famous Chinese forum social networking site http://movie.douban.com/. It contains user-user friendships and user-movie ratings, which is publicly available[3].
- *Epinions.* This dataset[4] is extracted from the consumer review website Epinions http://www.epinions.com/. The data also contains user-user trust relationships and numerical ratings.

Since ratings in *Ciao*, *Douban* and *Epinions* are all integers ranging from 1 to 5, we "binarize" them into boolean datasets: we consider items rated higher than 2 as consumed items. For DBLP, we use all citations occurring before year 2009 as the training set and leave all citations in or after 2009 for testing. For other datasets, we randomly choose 80% of each user's consumed items for training and leave the remainder for testing.

Methods Compared. The following eight recommendation methods, including six baselines, are tested.

- *TBPR-W.* Our TBPR model with weak ties ranked above strong ties (Equation 3).
- *TBPR-S.* Our TBPR model with strong ties ranked above weak ties (Equation 4).
- *BPR.* The classic method proposed in [42], coupled with matrix factorization for item scoring.

[3] https://www.cse.cuhk.edu.hk/irwin.king.new/pub/data/douban

[4] http://www.trustlet.org/wiki/Epinions_dataset

- *SBPR.* The Social BPR method proposed in [48], using the assumption that social items are ranked higher than non-social items.
- *SBPR-N.* A naive version of SBPR which ranks social items without considering the number of ties that consumed the items. Comparisons between SBPR-N and TBPR is to show that TBPR's improvement over SBPR is irrespective of whether the number of ties is considered or not.
- *Implicit MF (WRMF).* Weighted matrix factorization using a point-wise optimization strategy for implicit user-item feedback [24].
- *Random.* Randomly sample the non-consumed items to form a ranked list for each user.
- *Most Popular.* This is a non-personalized baseline which ranks all items based on their global popularity, i.e., the number of users that consumed an item.

All experiments are conducted on a platform with 2.3 GHz Intel Core i7 CPU and 16 GB 1600 MHz DDR3 memory. Grid search and 5-fold cross validation are used to find the best regularizer and we set $\lambda_u = \lambda_q = 0.01$ and $\lambda_\theta = 0.1$. The learning rate η of stochastic gradient descent is set to 0.1 for $\theta_{\mathcal{G}}$ and 0.01 for other parameters.

Evaluation Metrics. The following metrics are used to measure the prediction accuracy.

- *Recall@K* (Rec@K). This metric quantifies the fraction of consumed items that are in the top-K ranking list sorted by their estimated rankings. For each user u we define $S(K; u)$ as the set of already-consumed items in the test set that appear in the top-K list and $S(u)$ as the set of all items consumed by this user in the test set. Then, we have

$$\textit{Recall@K}(u) = \frac{|S(K; u)|}{|S(u)|}.$$

- *Precision@K* (Pre@K). This measures the fraction of the top-K items that are indeed consumed by the user (test set):

$$\textit{Precision@K}(u) = \frac{|S(K; u)|}{K}.$$

- *Area Under the Curve (AUC).*

$$AUC = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{E}_u|} \sum_{(i,j) \in \mathcal{E}_u} \delta((x_{ui} - x_{uj}) > 0),$$

where $\mathcal{E}_u = \{(i, j) | i \in S(u) \wedge j \in \mathcal{I} \setminus \mathcal{C}_u^{self}\}$ and $(x_{ui} - x_{uj}) > 0$ indicates that for user u, item i is ranked ahead of item j.

- *Mean Average Precision (MAP).* Let $C(u)$ be the set of user u's candidate items for ranking in the test set. The average precision for u is:

$$AP(u) = \frac{1}{|S(u)|} \sum_{K=1}^{|C(u)|} \textit{Precision@K}(u),$$

and the mean average precision will be:

$$MAP = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} AP(u).$$

- *Mean Reciprocal Rank (MRR).* Let $R(u)$ be the ranking of items in $C(u)$ in descending order, then for any item i in $S(u)$,

		Random	Popular	WRMF	BPR	SBPR-N	SBPR	TBPR-S	TBPR-W	Impv%
DBLP	Pre@5	0.000374	0.009642	0.029259	0.031081	0.031592	0.034522	0.033807	**0.040680**	17.8%†
	Rec@5	0.000362	0.004552	0.026533	0.029588	0.029229	0.030541	0.030146	**0.036152**	18.4%†
	AUC	0.491990	0.687722	0.823800	0.866391	0.863784	0.873853	0.867394	**0.903863**	3.43%†
	MAP	0.001235	0.006007	0.026071	0.030382	0.029903	0.031865	0.031434	**0.038393**	20.5%†
	NDCG	0.150952	0.171013	0.210249	0.224147	0.225434	0.231224	0.228612	**0.241568**	4.47%†
	MRR	0.003341	0.038732	0.088439	0.093422	0.092076	0.095209	0.094304	**0.106519**	11.9%†
Ciao	Pre@5	0.001068	0.008219	0.016689	0.015152	0.015954	0.016752	0.016614	**0.018497**	10.4%†
	Rec@5	0.000435	0.014629	0.019048	0.017522	0.017986	0.021813	0.021609	**0.023532**	7.88%†
	AUC	0.508182	0.671325	0.727143	0.770230	0.770574	0.775404	0.770964	**0.798210**	2.94%†
	MAP	0.001537	0.015193	0.017857	0.018991	0.018971	0.019704	0.019400	**0.024158**	22.6%†
	NDCG	0.125384	0.167200	0.171261	0.175962	0.179664	0.186527	0.185729	**0.199382**	6.89%†
	MRR	0.005500	0.033362	0.052541	0.050175	0.051811	0.054098	0.052668	**0.060842**	12.5%†
Douban	Pre@5	0.027144	0.137217	0.137844	0.154097	0.170087	0.170754	**0.211767**	0.170667	24.0%†
	Rec@5	0.005559	0.023895	0.040620	0.025105	0.029743	0.038584	**0.043969**	0.038549	8.24%†
	AUC	0.553510	0.839267	**0.974293**	0.971433	0.972291	0.972306	0.974195	0.972039	-0.01%‡
	MAP	0.014030	0.057165	0.072330	0.077991	0.073461	0.078055	**0.099851**	0.078199	27.9%†
	NDCG	0.299853	0.392563	0.385310	0.438571	0.434797	0.454351	**0.488024**	0.454644	7.41%†
	MRR	0.077733	0.289429	0.301047	0.300290	0.356630	0.357672	**0.437344**	0.356532	22.3%†
Epinions	Pre@5	0.000132	0.016230	0.021320	0.022658	0.023797	0.024945	0.024802	**0.026989**	8.19%†
	Rec@5	0.000040	0.014579	0.018795	0.020810	0.020552	0.021308	0.021819	**0.023960**	12.4%†
	AUC	0.514609	0.784285	0.890701	0.894476	0.894034	0.901279	0.901306	**0.918934**	1.96%†
	MAP	0.000703	0.012759	0.019503	0.021544	0.021663	0.022174	0.022452	**0.024461**	10.3%†
	NDCG	0.126838	0.174815	0.195235	0.198665	0.199031	0.209530	0.207757	**0.225629**	7.68%†
	MRR	0.001892	0.051963	0.067934	0.073059	0.073879	0.075097	0.073299	**0.086588**	15.3%†

Table 2: Performance evaluations on all users (boldface font denotes the winner in that row).

we denote its position in $R(u)$ as $rank_i^u$. Thus the mean reciprocal rank is computed as follows:

$$MRR = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \sum_{i=1}^{|S(u)|} \frac{1}{rank_i^u}.$$

- *Normalized Discounted Cumulative Gain (NDCG).* This is widely used in information retrieval and it measures the quality of ranking through discounted importance based on positions. In recommender systems, NDCG is computed as following:

$$NDCG = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{DCG_u}{IDCG_u},$$

where DCG and IDCG (Ideal Discounted Cumulative Gain) are in turn defined as:

$$DCG_u = \sum_{i \in S(u)} \frac{1}{\log_2(rank_i^u + 1)},$$

$$IDCG_u = \sum_{i=1}^{|S(u)|} \frac{1}{\log_2(i + 1)}.$$

6.2 Results and Analysis

Table 2 demonstrates the performance of all eight recommendation methods on all four datasets, measured by six different accuracy metrics. We also conduct a paired difference test (dependent t-test for paired samples) between TBPR (whichever version is better) and the best baseline over all six metrics on each dataset. In Table 2, † indicates that the result of a paired difference test is significant at $p < 0.05$ with degree of freedom as $\#users - 1$ on each dataset and ‡ indicates the result is not significant. Generally speaking, TBPR outperforms all six baselines in all but one cases and moreover, all the results in which TBPR outperforms the best baseline are statistically significant at $p < 0.05$.

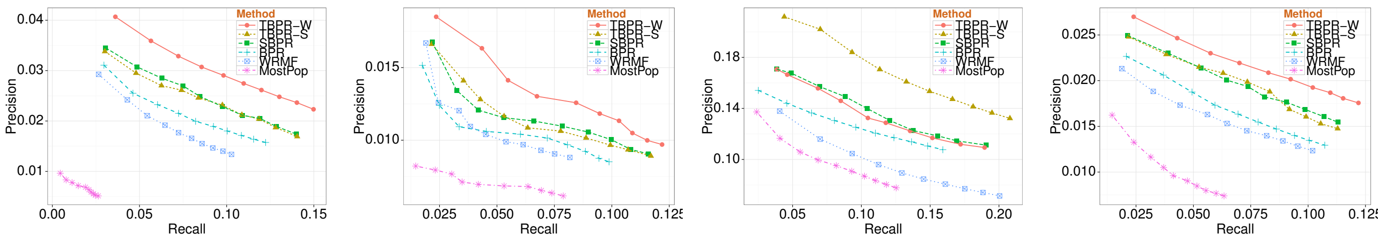
TBPR Models vs. Baselines. For the sake of clarity, in the last column of Table 2 we provide the relative improvement achieved by TBPR-W or TBPR-S (whichever is better) over the best baseline, determined on a row-by-row basis: E.g., for Pre@5 on Epinions, the best baseline is SBPR.

We observe that TBPR, with very few exceptions, outperforms the best baseline on all datasets and for all metrics. Considering the different metrics, the gap between TBPR and the baselines is typically larger for Rec@5, Pre@5, MAP, and MRR, while the smallest gaps are observed for AUC. BPR and SBPR are also quite strong in terms of AUC. This is due to a clear connection between optimizing AUC and the objective of BPR (and its extensions such as SBPR and our TBPR). For lack of space, we omit the details and refer the reader to [42].

In terms of datasets, the gap between TBPR and the baselines is generally larger on DBLP and Douban. For DBLP, there are four metrics (Pre@5, Rec@5, MAP, MRR) w.r.t. which TBPR's improvement is above 10%; For Douban, the advantage is more apparent: there are three metrics (Pre@5, MAP and MRR) w.r.t. which TBPR's improvement compared to the best baseline is 24.0%, 27.9% and 22.3%, respectively.

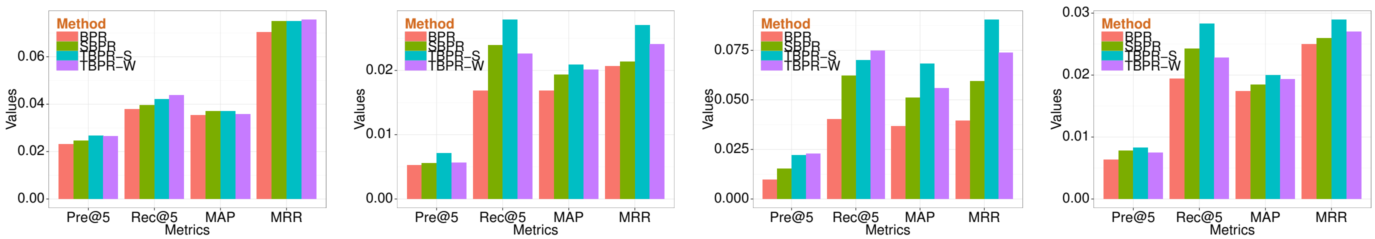
Note that although the two variants of TBPR assume reverse ordering between STC (Strong-Tie-Consumed) items and WTC (Weak-Tie-Consumed) items, they both outperform BPR. This may appear unintuitive, as one may imagine that if one particular ordering performs well, the reverse ordering should give inferior performance. To interpret these results, first recall that BPR only orders consumed items ahead of all non-consumed ones (including social and non-social), whereas both variants of TBPR order social items ahead of non-social items. The fact that both TBPR variants beat BPR actually further attests to the core intuition held by the large body of work on social recommendation: users tend to prefer social items to non-social items.

As to at least one variant of TBPR outperforming SBPR, recall



(a) DBLP (b) Ciao (c) Douban (d) Epinions

Figure 3: Precision@K vs Recall@K on all users, where K ranges from 5 to 50



(a) DBLP (b) Ciao (c) Douban (d) Epinions

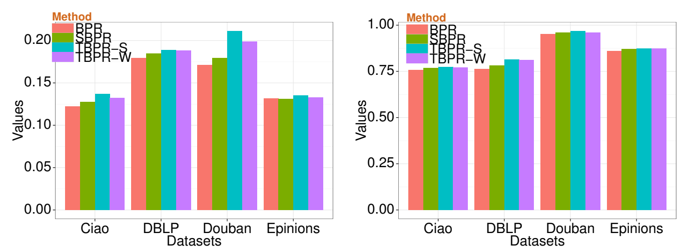
Figure 4: Performance evaluations on cold-start users (Recall, Precision, MAP, MRR)

that the key difference between TBPR and SBPR is the internal ordering amongst all social items of a user. SBPR "ranks" social items based on the number of ties that consumed the items, while the TBPR ordering is based on tie type. In fact, for any particular category of social items, e.g., WTC items, we do not impose any further internal ordering. This being the case, one may argue that the improvement of TBPR over SBPR might seem to lie in the fact that SBPR takes into account the number of ties and TBPR does not. Therefore we also implement a naive version of SBPR, which ranks social items without taking the number of ties that consumed the items into consideration. The comparisons demonstrate that both variants of TBPR outperform SBPR-N, suggesting that our idea of using tie type to categorize and rank social items is better.

TBPR-S vs. TBPR-W. We observe from Table 2 that TBPR-W beats TBPR-S on DBLP, Ciao, Epinions, while TBPR-S performs better on Douban. This indicates that on average users in different datasets may have different preferences over strong and weak ties, which further raises a question that do different users actually have distinct inherent biases toward STC items and WTC items? In fact, this leads to an interesting direction for future work, which is to personalize the ordering of STC and WTC items and learn it for each individual user. It is, however, an open question whether each user will have enough social items to allow the learning of a personalized-ordering model.

Recall and Precision. Figure 3 depicts Recall (X-axis) vs. Precision (Y-axis) achieved by six recommendation methods. We exclude Random since it is much worse than Most Popular. Data points from left to right on each line were calculated at different values of K, ranging from 5 to 50. Clearly, the closer the line is to the top right corner (of the plot area), the better the algorithm is: which indicates that both recall and precision are high. We can see that either TBPR-W or TBPR-S dominates all baselines, consistent with the findings in Table 2. In addition, the trade-off between recall and precision can be clearly observed from Figure 3.

Comparisons on Cold-Start Users. We further investigate the performance of various recommendation methods on *cold-start* users. As is common practice, we define users that consumed less than



(a) NDCG (b) AUC

Figure 5: NDCG and AUC comparisons on cold-start users

	Pre@5	Rec@5	AUC	MAP	NDCG	MRR
DBLP	8.53%	10.8%	4.27%	0.14%	2.26%	0.88%
Ciao	28.0%	16.7%	0.73%	8.19%	7.69%	26.3%
Douban	49.6%	20.0%	0.80%	33.9%	17.6%	52.3%
Epinions	6.39%	16.8%	0.30%	8.61%	2.77%	11.6%

Table 3: Percentage improvement of TBPR (the better of TBRP-W and TBPR-S) over the best basline on cold-start users

five items as cold-start. Table 3 demonstrates the percentage improvement of TBPR (the better of TBPR-S and TBPR-W) over the best baseline. By comparing Tables 2 and 3, we can see that more often than not, the improvement by TBPR is larger for cold-start users. For instance, on Ciao and Douban, the improvement is larger w.r.t. five out of all six metrics.

We further compare BPR, SBPR, and TBPR on all six metrics in Figures 4 and 5. SBPR outperforms BPR in all cases, which again confirms the benefit of taking social network information into consideration for recommender systems. Note that in most cases, TBPR-S is slightly better than TBPR-W. This is reasonable as cold-start users may first rely on strong ties who are more trust-worthy to them.

Finally, from our comprehensive experiments, it is fair to conclude that both TBPR-W and TBPR-S are effective social recommendation methods based on their convincing performance on not only all users, but also cold-start users.

7. CONCLUSIONS AND FUTURE WORK

In this work, we present a new social recommendation method for implicit feedback data. Motivated by the seminal work in sociology by Granovetter [21, 22], we recognize the effects of strong and weak ties, in particular, the role played by weak ties in spreading novel information over social networks. Our model is a non-trivial extension to the Bayesian Personalized Ranking (BPR) model that is aware of the important distinction between strong and weak ties in social networks. We categorize "social items" (i.e., those not consumed by a user herself, but were consumed by the user's social ties) into three groups, depending on whether an item was consumed by the user's strong ties, weak ties, or both. We propose to use Jaccard's coefficient to compute tie strengths in a given social network, and then devise an EM-style algorithm that is capable of simultaneously learning the tie strength threshold and the latent feature vectors of all users and items. Our comprehensive experimental results on four real-world datasets clearly demonstrate the efficacy of our proposed methods and their superiority over existing pairwise recommendation models such as BPR [42] and SBPR [48], as well as point-wise ones such as WRMF [24].

This work opens up plenty of opportunities for future research. First, as pointed out in Section 6.2, we conjecture that an even more personalized TBPR model warrants careful considerations, since it is plausible that while some users prefer items consumed by weak ties over those by strong ties, other users may behave in the opposite way. Also, it is interesting to learn personalized tie strength thresholds. Namely, the model may assume each user is associated with a different threshold for classifying strong and weak ties.

Another interesting future direction is to consider other node-similarity metrics such as Adamic-Adar or Katz score, which can be used in lieu of Jaccard for computing tie strengths: a comprehensive empirical comparison on various tie strength definitions is worthwhile. Last but not the least, one may couple the TBPR model with other item scorers like kNN, instead of matrix factorization.

ACKNOWLEDGEMENT

This work is supported by Natural Science Foundation of Zhejiang Province (Grant Number: LZ13F020001).

8. REFERENCES

[1] Y. Wu *et al.,* Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In *WSDM*, pages 153–162, 2016.

[2] J. Li *et al.,* Robust Unsupervised Feature Selection on Networked Data. In *SIAM International Conference on Data Mining*, 2016.

[3] X. Wang *et al.,* Recommending Groups to Users Using User-Group Engagement and Time-Dependent Matrix Factorization. In *AAAI*, 2016.

[4] I. Kahanda and J. Neville. Using Transactional Information to Predict Link Strength in Online Social Networks. In *ICWSM*, pages 74–81, 2009.

[5] R. Xiang, J. Neville and M. Rogati. Modeling relationship strength in online social networks. In *WWW*, pages 981–990, 2010.

[6] V. Arnaboldi, A. Guazzini and A. Passarella. Egocentric online social networks: Analysis of key features and prediction of tie strength in Facebook. *Computer Communications*, 36(10):1130–1144, Elsevier, 2013.

[7] E. Gilbert. Predicting tie strength in a new medium. In *CSCW*, pages 1047–1056, 2012.

[8] A. Petróczi, T. Nepusz and F. Bazsó. Measuring tie-strength in virtual social networks. *Connections*, 27(2):39–52, 2007.

[9] R. Reagans. Preferences, identity, and competition: Predicting tie strength from demographic data. *Management Science*, 51(9):1374-1383, INFORMS, 2005.

[10] A. Kavanaugh *et al.,* Weak ties in networked communities. *The Information Society*, 21(2):119-131, Taylor & Francis, 2005.

[11] N. Christakis and J. Fowler. Connected: how your friends' friends' friends affect everything you feel, think, and do. *New York, NY: Little, Brown, and Company*, 2009.

[12] L. Gee *et al.,* Social Networks and Labor Markets: How Strong Ties Relate to Job Finding On Facebook's Social Network. *Journal of Labor Economics*, 2016.

[13] C. Zhang *et al.,* Content+ attributes: A latent factor model for recommending scientific papers in heterogeneous academic networks. *Advances in Information Retrieval*, pages 39–50, Springer, 2014.

[14] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[15] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, 2005.

[16] A. Das *et al.,* Google news personalization: Scalable online collaborative filtering. In *WWW*, pages 271–280, 2007.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.

[18] D. Easley and J. M. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[19] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[20] E. Gilbert and K. Karahalios. Predicting tie strength with social media. In *CHI*, pages 211–220. ACM, 2009.

[21] M. S. Granovetter. The strength of weak ties. *American journal of sociology*, pages 1360–1380, 1973.

[22] M. S. Granovetter. *Getting a job: A study of contacts and careers*. University of Chicago Press, 1974.

[23] G. Guo, J. Zhang, and N. Yorke-Smith. A novel bayesian similarity measure for recommender systems. In *IJCAI*, 2013.

[24] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

[25] M. Jamali and M. Ester. TrustWalker: a random walk model for combining trust-based and item-based recommendation. In *KDD*, pages 397–406, 2009.

[26] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, pages 135–142, 2010.

[27] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39 – 43, March 1953.

[28] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434, 2008.

[29] Y. Koren *et al.* Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[30] A. Krohn-Grimberghe *et al.,* Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *WSDM*, pages 173–182. ACM, 2012.

[31] D. Liben-Nowell and J. M. Kleinberg. The link prediction problem for social networks. In *CIKM*, pages 556–559, 2013.

[32] W. Lu *et al.,* Optimal recommendations under attraction, aversion, and social influence. In *KDD*, pages 811–820, 2014.

[33] H. Ma, I. King, and M. R. Lyu. Learning to recommend with explicit and implicit social relations. *ACM TIST*, 2(3):29, 2011.

[34] H. Ma *et al.,* SoRec: social recommendation using probabilistic matrix factorization. In *CIKM*, 2008.

[35] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, pages 287–296, 2011.

[36] D. Oard and J. Kim. Implicit feedback for recommender systems. In *AAAI Workshop on Recommender Systems*, pages 81–83, 1998.

[37] J.-P. Onnela *et al.*, Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences*, 104(18):7332–7336, 2007.

[38] R. Pan *et al.,* One-class collaborative filtering. In *ICDM*, pages 502–511, 2008.

[39] W. Pan and L. Chen. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. In *IJCAI*, pages 2691–2697, 2013.

[40] K. Panovich, R. Miller, and D. Karger. Tie strength in question & answer on social network sites. In *CSCW*, pages 1057–1066, 2012.

[41] S. Rendle and C. Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *WSDM*, 2014.

[42] S. Rendle *et al.,* BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[43] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[44] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2007.

[45] A. Wu *et al.,* Detecting professional versus personal closeness using an enterprise social network site. In *CHI*, 2010.

[46] S.-H. Yang *et al.,* Like like alike: joint friendship and interest propagation in social networks. In *WWW*, 2011.

[47] M. Ye *et al.,* Exploring social influence for recommendation: a generative model approach. In *SIGIR*, 2012.

[48] T. Zhao, J. McAuley, and I. King. Leveraging social connections to improve personalized ranking for collaborative filtering. In *CIKM*, pages 261–270, 2014.