Predicting Item Adoption Using Social Correlation

Freddy Chong Tat Chua* Hady W. Lauw† Ee-Peng Lim*

Abstract

Users face a dazzling array of choices on the Web when it comes to choosing which product to buy, which video to watch, etc. The trend of social information processing means users increasingly rely not only on their own preferences, but also on friends when making various adoption decisions. In this paper, we investigate the effects of social correlation on users' adoption of items. Given a user-user social graph and an item-user adoption graph, we seek to answer the following questions: 1) whether the items adopted by a user correlate to items adopted by her friends, and 2) how to incorporate social correlation in order to improve prediction of unobserved item adoptions. We propose the *Social Correlation* model based on *Latent Dirichlet Allocation (LDA)* that decomposes the adoption graph into a set of latent factors reflecting user preferences, and a social correlation matrix reflecting the degree of correlation from one user to another. This matrix is learned (rather than pre-assigned), has probabilistic interpretation, and preserves the underlying social network structure. We further devise a *Hybrid* model that combines a user's own latent factors with her friends' for adoption prediction. Experiments on Epinions and LiveJournal data sets show that our proposed models outperform the approach based on latent factors only (LDA).

1 Introduction

Unprecedented progress and innovation provide consumers a wide variety of choices. Consumer items such as books, cameras and movies come in various subjects, features and genres. Online shopping provides access to these items to anyone with an internet connection. Consequently, sellers anywhere can reach consumers anywhere, and consumers have access to increasing number of products. The direct effect is consumers have a harder time making purchasing decisions, while sellers do not know what to sell and whom to sell it to. Beyond commerce, users face a similar problem on the Web in general, when deciding which article to read, which group to join, etc.

To address this information overload, retailers attempt to assist consumers by putting in place decision-making aids such as bestseller lists, listing items frequently bought together, etc. However, given the limited space in bestseller lists or any recommendation list targeted at everyone, such aids would favor the very popular items. Some merchants, such as Amazon and Netflix, have put in place more personalized recommender systems based on the individual user's past transactions. However, such approaches frequently suffer from the cold start problem: no recommendation can be generated for users who have purchased very few items. Therefore, while attractive retail opportunity lies in the long-tail products, it is difficult for such products to be matched to the relevant users.

In a trend known as social information processing, users increasingly rely on one another to organize the complex information on the Web. This is evident from the abundant amount of user-generated content, such as tags, ratings, and reviews, all of which collectively aim to allow items to be more easily discovered by other users. Social networks have also become a conduit for discovering relevant information. In such platforms as Twitter or Epinions, users can opt to receive only content generated by other users whom they follow or trust. A user's choices are increasingly driven not only by personal preferences, but also by the preferences of others in their social networks. This gives rise to the phenomenon of *social correlation*, whereby users who are socially related tend to make similar choices.

In this paper, we therefore aim to address the item adoption prediction problem by studying how social correlation plays a role in user adoption of items. Here, item adoption could refer to various actions such as buying a product, writing a product review, joining a group, etc. We model the adoption relationship between users and items as an undirected bipartite *adoption graph* $\mathcal{G}_a(V, U, E)$ where V represents a set of items, U represents a set of users and E represents the undirected adoption links between V and U. We also assume as input a *social graph* $\mathcal{G}_s(U, F)$, where U represents the same set of users as in $\mathcal{G}_a$ and F represents the social links between users. An edge exists from u_1 to u_2 if u_1 befriends, trusts, or follows u_2. In both $\mathcal{G}_a$ and $\mathcal{G}_s$, we only require the binary expression of the links (present

*Singapore Management University

†Institute for Infocomm Research

or absent), and do not use any other form of information such as ratings or review text to keep our model simple and general.

Given $\mathcal{G}_a$ and $\mathcal{G}_s$, we seek to address the following problems:

- *Learning the extent to which a user relies on social correlation, as opposed to her personal preferences, in making adoption choices.* For a given social link $(u_1, u_2) \in F$, we would like to learn a weight that reflects the extent to which u_1's latent factors correlate with the latent factors of u_2.

- *Predicting the items that a user is likely to adopt based on social correlation.* For a given pair of user u and item v, we would like to learn the probability that an adoption link (u, v) would exist in E.

Factorization-based approaches can model a user's personal preferences [14]. One such factorization is Latent Dirichlet Allocation (LDA) [4], which learns a set of latent factors by factorizing the adjacency matrix of the adoption graph into two matrices: one that reflects the importance of each latent factor to users, and another that does the same for items. However, this approach is inadequate because it assumes that all items adopted by a user can all be explained by the user's and items' latent factors.

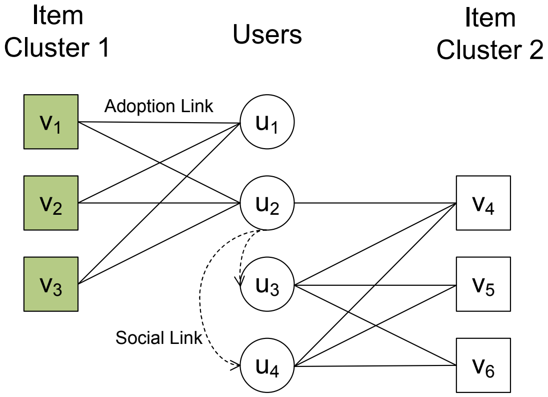

Figure 1: Example Scenario of Adoption (solid) and Social Links (dotted)

Consider the example scenario in Figure 1. There are two clusters of items: $\{v_1, v_2, v_3\}$ and $\{v_4, v_5, v_6\}$. Suppose that each cluster groups together items with similar latent factors. Users u_1 and u_2 have similar preferences, adopting items in the first cluster. Users u_3 and u_4 adopt items in the second cluster. Given that items in a cluster share similar latent factors, these adoptions can largely be explained by the users' having similar latent factors. However, u_2's adoption of v_4 cannot be clearly explained by latent factors alone. Taking into account u_2's social links (dotted lines) to u_3 and u_4, we hypothesize that in the case of v_4, u_2 depends on the preferences of her friends u_3 and u_4.

We propose to model social correlation directly using factorization-based approaches. Some users may primarily rely only on their own latent factors in making adoptions. We say that these users have high *self-dependency* values. However, most users rely on a mixture of self-dependency and social correlation. This is modeled by a user-user *social correlation matrix* I. Based on a generative model, our approach assumes that a user u_1 adopts an item based on her preferences on latent factors of the item with a probability proportional to $i_{u_1,u_1} \in I$ also known as *Self-Dependency*, and based on another user u_2's latent factors with probability proportional to $i_{u_1,u_2} \in I$. Here, $\sum_u i_{u_1,u} = 1$. Hence, we seek to learn both a user's latent factors and the social correlation matrix from the given adoption and social graphs.

We make the following contributions in this paper:

1. We propose two factorization models that we call the Social Correlation and Hybrid models. Social Correlation model decomposes an adoption graph and social graph into three components: users' latent factors, items' latent factors, and social correlation. While Hybrid model combines the merit of the Social Correlation model and LDA.

2. Our proposed models derive the social correlation weights from the factorization process, instead of relying on a social graph with pre-assigned link weights. In some cases, the weights are not known before hand. Even if the social graph comes with some form of weights (e.g., friendship strength), they may not accurately reflect the dependency and correlation among users.

3. To evaluate our proposed models, we conduct comprehensive experiments on two real-life data sets from Epinions and LiveJournal. The results show that our proposed models outperform the approach relying on latent factors alone. We also show that the Hybrid outperforms Social Correlation.

The rest of the paper is organized as follows. Section 2 will discuss the past research done on modeling items and users relationship. We establish the existence of correlation between adoption and social links in Section 3 through hypothesis testing. In Section 4, we apply Latent Dirichlet Allocation (LDA) to model user adoption of items based on latent factors. In Section 5, we incorporate social correlation into the factorization model. We then proceed to evaluate our methods in Section 6. Finally we conclude our paper in Section 7.

2 Related Work

2.1 Social Correlation Here, we review several concepts related to social correlation, such as homophily, influence, k-exposure, etc. Notably, we go beyond just establishing or measuring social correlation, to also make use of it for adoption prediction.

Fond and Neville [11,20] established that social correlation was a result of two processes that happen alternatively over a period of time: "homophily" causing users with similar attributes to form social links, and "influence" causing users with social links to become more similar in attributes. The notion of homophily is a well known phenomenon in sociology. McPherson et al. [19] surveyed articles establishing that homophily exists in various social contexts such as marriage, friendship, co-workers, classmates, involving similarity factors such as socio-demographic attributes. Singla and Richardson [22] also established the correlation of search queries among instant messaging friends. In our work, we are concerned only with the existence of social correlation and its use for adoption prediction, and not with the underlying causes (homophily vs. influence), which are not always observable from the data.

Liu et al. sought to measure influence [16] based on clearly observable "following" behaviors. For instance, it looked at how Twitter users re-tweeted postings by others, or how authors published papers on the same topics as cited papers. They first obtain the topic distribution of every author based on the papers they wrote. Then for each author a, they decide who influences a based on the latent factors of the authors whom a cited from. Our work is different in the following ways. First, our focus is the adoption prediction problem, while their focus is on measuring influence and how it varies with the various number of hops in the social graph. Second, our model assumes that any friend (and not just certain friends e.g., authors cited) could be influencers. For example, a user who buys an item does not explicitly state whom she bases her decision on. In such cases, the possible number of influencers can be very large and their method may not scale up. Therefore, we view our approach as a more generalized approach that can work in generic settings.

Also related is the notion of k-exposure: the likelihood that a user would adopt an item increases with the number k of her friends who have adopted it. Several works have studied k-exposure with respect to such adoptions as choosing which Wikipedia article to edit or which LiveJournal community to join [2, 6, 7]. The fundamental assumption here is that every user is correlated with their friends in the same way. All that matters is the number of friends who have adopted an item. In contrast, we do not make the same assumption. In our approach, a user may be correlated with each friend differently, and may have different self-dependency values.

Ma et al. extended the Bayesian Probabilistic Matrix Factorization (BPMF) models for rating prediction by adding social factors [17, 18]. They used the latent factors of users and items learned from BPMF coupled with the weighted values of the social links for item ratings prediction. Importantly, they assume the existence of the weighted values that reflect the relationship strength between each pair of friends. In the absence of known weights, all users may be weighted equally. In this work, we do not make the same assumption, and show that it is possible to learn these weighted values through an optimization process.

Some prior work focused on how influence propagated across a network. Assuming a propagation framework such that an adoption by a user would probabilistically trigger a similar adoption by her friends, an influential user is one whose initial adoption would eventually result in the most number of total adoptions by all users [13]. The problem of influence maximization is orthogonal to our problem, in that influence maximization is more concerned with the total number of adoptions triggered, while we are concerned more with predicting individual adoption cases.

Influence is also addressed by Yang and Leskovec as a form of information diffusion [23] with temporal dynamics. However, their notion of influence requires the explicit adoption of item while we consider in terms of latent factors.

2.2 Factorization The Bayesian Probabilistic Matrix Factorization (BPMF) is a popular model for low rank matrix approximation [21] method by Salakhutdinov. The model avoids overfitting of other methods such as SVD by adding Gaussian noise to the sparse data. The Gaussian noise acts as a regularizer to avoid overfitting the factorized matrices to the sparse data. Salakhutdinov then showed that the model can be approximated using a Gibbs Sampling method. The BPMF method subsequently was applied by Koren to rating prediction in the Netflix Prize Competition [14]. Koren combined the generalization properties of latent factor models to neighborhood methods in collaborative filtering. Koren also extended the factor models to modeling temporal dynamics [15].

When modeling ratings, it is appropriate to use BPMF because rating scores can be approximated to follow the Gaussian distribution. When we want to model simpler discrete relationships, the Latent Dirichlet Allocation (LDA) is more suitable [4]. Instead of Gaussian noise as regularizers, the LDA uses Dirichlet distributions as smoothing priors which essentially behaves in the same way as regularizers.

There are existing works that uses Dirichlet distributions to model item - user and user - user relationships. Balasubramanya and Cohen had proposed Block-LDA for modeling protein interactions [3]. The Block-LDA tries to unite the Mixed Membership Stochastic Blockmodels [1] and LDA to jointly model the relationships. However, their approach and assumptions are currently restricted to protein interactions only.

3 Correlation of Social & Adoption Links

We justify our research motivation by first establishing that a correlation exists between social and adoption links, i.e., whether users with social links also tend to share common adoptions. We do this by performing hypothesis tests on a real world data set obtained from Epinions, a product review site. The social graph in Epinions consists of trust links formed when a user indicates her trust on another user. These trust links are directional and not necessarily reciprocal. An adoption link exists between a user and an item (product) if the user has written a review for the item.

We collected the data set by crawling the Epinions site, focusing only on the *Videos & DVDs* category. The size of the data set is given in Table 1. In total, there are close to 40K users and 7K items. There are also more than 300K social links and 80K adoption links. Both social and adoption links are binary (0 or 1). Although the adoption links are binary in Epinions data set, we can also handle weighed adoption links that represents adoption of the same item multiple times.

Table 1: Epinions: Data Size

	Count
no of users $\lvert U\rvert$	39,946
no of items $\lvert V\rvert$	6,949
no of adoption links $\lvert E\rvert$	83,763
no of social links $\lvert F\rvert$	331,509

We perform hypothesis testing using the Fisher Exact Test [10]. Our null hypothesis H_0 states that the probability of two users having a common adoption is independent of whether the two users have a trust link between them. Rejecting the null hypothesis implies accepting the alternate hypothesis H_1, which states that the probability of common adoption is dependent on having social link.

We perform the Fisher Exact Test on the contingency table in Table 2. Each value in the table represents the number of user pairs for a combination of social link and common item adoption scenarios. The numbers in parentheses are the expected values if the social graph is independent of the adoption graph. As shown in the table, the observed number of pairs with both common adoption and social link 24,197 is far greater than the expected 2,594.

Table 2: Epinions : Contingency Table For Pair of Users with Social and Adoption Links

	No Common Adoption	Has Common Adoption	Total
No Social Link	791,271,379 (791,249,776)	6,218,597 (6,240,200)	797,489,976
Has Social Link	307,312 (328,915)	24,197 (2,594)	331,509
Total	791,578,691	6,242,794	797,821,485

Using Fisher Exact Test, we obtain a p-value $< 2.2 \times 10^{-16}$ which indicates that we can reject H_0, and conclude that the presence of social links is correlated with the presence of adoption links. We also established similar conclusions on a second data set obtained from LiveJournal, but do not reproduce them here due to space consideration.

4 Factorization based on LDA

Our proposed approach is to first factorize the observed adoption graph E into user and item latent factors based on Latent Dirichlet Allocation (LDA), before learning the social correlation matrix I. In this section, we describe how we apply LDA for the item adoption prediction problem.

LDA was formerly conceived as a way of modeling unigram words in a document corpus [4]. Each document is seen as a collection of words and the words are generated as a result of the topics each document contains. Using documents and words as analogy, we view users in the adoption graph as documents, the items they adopt as words and the latent factors of the items as topics. We now express a statistical formulation of LDA, and give an alternative linear algebraic formulation later in this section.

The user u latent factor distribution θ_u follows a symmetric Dirichlet distribution with hyper-parameters ν, as follows:

$$\theta_u \sim Dirichlet(\nu)$$

The latent factor $z_{v,u} \in \{1, \ldots, T\}$ of each item v that the user u adopts is generated by the multinomial distribution with parameters θ_u, as follows:

$$z_{v,u} \sim Multinomial(\theta_u)$$

The item v that the user u will adopt is generated by the latent factor $z_{v,u}$ and the latent factor-item distribution β, as follows:

$$e_{v,u} \sim \beta | z_{v,u}$$

The latent factor-item distribution β follows a symmetric Dirichlet distribution with hyper-parameters ϕ, as follows:

$$\beta | z_{v,u} \sim Dirichlet(\phi)$$

In the alternative linear algebraic formulation, LDA is a factorization algorithm that takes as input an $M \times N$ matrix E and outputs a $M \times T$ matrix β and a $T \times N$ matrix θ. Here, T is the number of latent factors, M is the number of items, and N is the number of users. Intuitively, β represents the latent factors of items, and θ the latent factors of users.

Suppose our matrix E is as follows,

$$E = \begin{bmatrix} e_{v_1,u_1} & \cdots & e_{v_1,u_N} \\ \vdots & \ddots & \vdots \\ e_{v_M,u_1} & \cdots & e_{v_M,u_N} \end{bmatrix} \tag{4.1}$$

The LDA algorithm takes E as input and outputs β and θ.

(4.2)

$$LDA(E) = \begin{bmatrix} \beta_{v_1|1} & \dots & \beta_{v_1|T} \\ \vdots & \ddots & \vdots \\ \beta_{v_M|1} & \dots & \beta_{v_M|T} \end{bmatrix} \begin{bmatrix} \theta_{u_1,1} & \dots & \theta_{u_N,1} \\ \vdots & \ddots & \vdots \\ \theta_{u_1,T} & \dots & \theta_{u_N,T} \end{bmatrix}$$

where each column in β and θ sums to 1. Solving for these two matrices is fundamentally a likelihood optimization problem subjected to the probability constraints. Blei showed that the matrices are learned using variational expectation maximization [4]. Griffiths and Steyvers subsequently showed that LDA can be learned easily using Gibbs Sampling [12].

When we multiply the matrices β and θ, we obtain the dense matrix E' which gives us the probability whether the links exist in the original sparse matrix E. As shown in Equation 4.3, E' is an approximation of the original E, only denser because it also produces probability values for the unobserved links in E.

$$E \approx (E' = \beta\ \theta) \tag{4.3}$$

As the number of latent factors T approaches a larger value, the product of the factorized matrices E' gets more and more similar to E. However, this is not desirable because we lose the generalization properties of factorization algorithms and the solution becomes more over-fitting to E.

5 Factorization with Social Correlation

Factorization by LDA alone is not sufficient to model user adoption of items as it does not account for the social correlation effect.

Social Correlation Matrix; We propose a $N \times N$ *social correlation* matrix I to tell us how likely it is that a user will adopt an item based on the latent factors of other users. Each element $i_{u,u'}$ reflects the likelihood that the user u will be correlated to u', in the sense of making adoption decision based on the latent factors of u'. $i_{u,u}$ is the **self-dependency** of user u, or the likelihood that u relies on her own latent factors. The social correlation matrix is derived as:

$$E \approx E' I^T \tag{5.4}$$

To properly reflect the notion of correlation, I cannot just be any $N \times N$ matrix. We require that I must have the following properties:

- *It is probabilistic.* Each element $i_{u,u'}$ is in the range of [0, 1]. For each user u, we also have $\sum_{u'} i_{u,u'} = 1$.
- *It preserves the social network structure.* Since social correlation is based on the underlying social network structure, $i_{u,u'}$ should have non-zero value only if there is a social link from u to u', i.e., $i_{u,u'} > 0 \Rightarrow (u, u') \in F$. In addition, we also learn the self-dependency values $i_{u,u}$ for each user u.

I can be obtained in several ways. The naive way is to calculate I by multiplying E with the inverse of E', i.e. $I = (E')^{-1}E$. This naive way will not work for several reasons. First, I may over-fit leading to poor results in link prediction. The obtained $E'I^T$ will be as sparse as E, and thus the factorization does not help in link prediction. Second, I may have values outside the range of $[0, 1]$. In fact, they may range from negative infinity to positive infinity. Such values do not have clear semantics and it is hard to interpret the meaning of these values. Third, I may have non-zero values even if the users are not connected by social links.

Hence, instead of obtaining an exact I, we will obtain an approximated I such that we minimize the error $|E - E'I^T|$, subject to the above-mentioned constraints (probabilistic, social network structure). To learn I with clear semantics, we formulate a statistical learning problem where the goal is to learn the I which maximizes the likelihood of observing the values in E. Maximizing the likelihood is the dual equivalent problem of minimizing error.

Since the graphs are sparse, algorithms that scale with the number of observed links would run faster. In the following, we formulate such an algorithm, and show that the complexity is indeed polynomial to the number of observed links.

Models. Once the social correlation matrix I has been learned, we can instantiate two adoption prediction models as follows.

- *Social Correlation* represents the approach of relying only on social correlation for item adoption. We compute $E'I^T$ (see Equation 5.4) based on the learned I, taking into account only the non-diagonal values of I, i.e., setting $i_{u,u} = 0, \forall u \in U$.
- *Hybrid* represents the approach of combining Social Correlation and LDA, by computing $E'I^T$ with the original learned I (with diagonal values retained).

We will experimentally establish the merits of these models with respect to LDA in Section 6.

Special Case. Our proposed formulation subsumes the underlying latent factors model. In the case where I is the identity matrix, with 1's as diagonal values and 0's otherwise, then $E'I^T$ degenerates to E', which is the outcome by LDA factorization.

5.1 Solution Formulation

We would like to illustrate the formulation of our algorithm using probabilistic explanations. Given a user u, we will like to know the probability that she will adopt the item v, given the user latent factors θ_u and the topic latent factors β.

Suppose now that we have the edges of the social graph F and the latent factors of all other users U including herself, we hypothesize that the user u adopts items based on the latent factor preferences of her friends and the user herself. We may restate the equation as follows,

$$P(e_{v,u}|\theta, \beta, F)$$

$$= \sum_{u' \in U} P(e_{v,u'}, f_{u,u'}|\theta, \beta, F) \tag{5.5}$$

$$= \sum_{u' \in U} P(e_{v,u'}|\theta, \beta) P(f_{u,u'}|F) \tag{5.6}$$

where $f_{u,u'}$ represents that u has a directed social link to u'. Also note that $e_{v,u}$ has become $e_{v,u'}$ on the right hand side of the equations. $P(f_{u,u'}|F)$ is either 0 or 1 since we do not model the probability of social links.

Equation 5.6 however is not a valid probability equation because it does not sum to 1. In fact, the values will exceed 1 due to the outer summation over u'. The reason is besides knowing the probability that u indicates u' as a friend in the social graph $P(f_{u,u'}|F)$ and the probability that u' adopts item v in the adoption graph $P(e_{v,u'}|\theta, \beta)$, we also need an additional component that tells us the probability that u depends on u' in the adoption graph $P(x_{v,u} = u'|I)$ (to be defined shortly). This additional component is the social correlation that we want to determine.

Hence, our proposed factorization model is to introduce the latent variable $x_{v,u}$ which tells us which u' that u depends on, and the social correlation I where its elements $i_{u,u'}$ gives us the probability that u follows the latent factors of u'. The special case is $u' = u$ which tells us the self-dependency of u. The higher is $i_{u,u}$, the less the user u depends on social correlation.

Putting the above intuition formally, the probability that u adopts an item v based on the social correlation I is given by:

$$P(e_{v,u}|\theta, \beta, F, I)$$

$$= \sum_{u'} P(e_{v,u'}, x_{v,u} = u', f_{u,u'}|\theta, \beta, F, I) \tag{5.7}$$

$$= \sum_{u'} P(e_{v,u'}|\theta, \beta) P(f_{u,u'}|F) P(x_{v,u} = u'|I) \tag{5.8}$$

For simplicity in the following derivations, we will take,

$$P(e_{v,u'}|\theta, \beta) = e'_{v,u'} \tag{5.9}$$

$$P(f_{u,u'}|F) = f_{u,u'} \tag{5.10}$$

$$P(x_{v,u} = u'|I) = i_{u,u'} \tag{5.11}$$

$$P(e_{v,u}|\theta, \beta, F, I) = \sum_{u'} e'_{v,u'} f_{u,u'} i_{u,u'} \tag{5.12}$$

To learn the social correlation values, we maximize the log likelihood of $e_{v,u}, \forall v \in V, \forall u \in U$, using the Expectation Maximization (EM) algorithm [9],

$$P(E|\theta, \beta, F, I) = \prod_{v,u} P(e_{v,u}|\theta, \beta, F, I) \tag{5.13}$$

$$\log P(E|\theta, \beta, F, I) = \sum_{v,u} \log P(e_{v,u}|\theta, \beta, F, I) \tag{5.14}$$

$$= \sum_{v,u} \log \sum_{u'} e'_{v,u'} f_{u,u'} i_{u,u'} \tag{5.15}$$

5.2 Expectation Maximization Algorithm

We first show the E Step. The E Step of the EM algorithm tries to infer for the latent variables using initial values of I,

$$P(x_{v,u} = u'|e_{v,u}, f_{u,u'}, \theta, \beta, F, I)$$

$$= \frac{P(x_{v,u} = u', e_{v,u'}, f_{u,u'}|\theta, \beta, F, I)}{\sum_{u''} P(x_{v,u} = u'', e_{v,u''}, f_{u,u''}|\theta, \beta, F, I)}$$

$$= \frac{P(e_{v,u'}|\theta, \beta) P(f_{u,u'}|F) P(x_{v,u} = u'|I)}{\sum_{u''} P(e_{v,u''}|\theta, \beta) P(f_{u,u''}|F) P(x_{v,u} = u''|I)}$$

(5.16)

$$= \frac{e'_{v,u'} f_{u,u'} i_{u,u'}}{\sum_{u''} e'_{v,u''} f_{u,u''} i_{u,u''}}$$

(5.17)

$$= h(u, u', v)$$

Since we have introduced $i_{u,u'}$ as a probabilistic weight, hence, it must sum to one.

$$\sum_{u'} i_{u,u'} = 1, \qquad \forall u \in U$$

Now, we aim to maximize the log likelihood with respect to the unknown social correlation I, subject to the above constraints. In order to include the constraints as part of the objective function, we introduce the Lagrange multipliers λ_u [5] and proceed to solve the following using differentiation,

$$\frac{d}{d\, i_{u,u'}}\Bigg[\sum_{v\in V}\Bigg(\sum_{u_0\in U}\log\Big(\sum_{u_1\in U} e'_{v,u_1} f_{u_0,u_1} i_{u_0,u_1}\Big) - \lambda_{u_0}\Big(\sum_{u_1\in U} i_{u_0,u_1} - 1\Big)\Bigg)\Bigg]$$

$$= \sum_{v\in V}\frac{e'_{v,u'}f_{u,u'}}{\sum_{u_1\in U} e'_{v,u_1}f_{u,u_1}i_{u,u_1}} - \lambda_u$$

To solve for λ_u, we equate the equation to 0 as follows,

$$\sum_{v\in V}\frac{e'_{v,u'}f_{u,u'}}{\sum_{u_1\in U} e'_{v,u_1}f_{u,u_1}i_{u,u_1}} - \lambda_u = 0$$

$$\lambda_u = \sum_{v\in V}\frac{e'_{v,u'}f_{u,u'}}{\sum_{u_1\in U} e'_{v,u_1}f_{u,u_1}i_{u,u_1}}$$

$$\lambda_u i_{u,u'} = \sum_{v\in V}\frac{e'_{v,u'}f_{u,u'}i_{u,u'}}{\sum_{u_1\in U} e'_{v,u_1}f_{u,u_1}i_{u,u_1}}$$

$$\lambda_u \sum_{u'\in U} i_{u,u'} = \sum_{v\in V}\frac{\sum_{u'\in U} e'_{v,u'}f_{u,u'}i_{u,u'}}{\sum_{u_1\in U} e'_{v,u_1}f_{u,u_1}i_{u,u_1}}$$

$$\lambda_u = \sum_{v} 1$$

It can be seen clearly from the equations above that λ_u is the number of items u has been observed to adopt. Now to solve for $i_{u,u'}$,

$$\lambda_u i_{u,u'} = \sum_{v\in V}\frac{e'_{v,u'}f_{u,u'}i_{u,u'}}{\sum_{u_1} e'_{v,u_1}f_{u,u_1}i_{u,u_1}} \tag{5.18}$$

$$i_{u,u'} = \frac{1}{\lambda_u}\sum_{v\in V}\frac{e'_{v,u'}f_{u,u'}i_{u,u'}}{\sum_{u_1} e'_{v,u_1}f_{u,u_1}i_{u,u_1}} \tag{5.19}$$

Recall in our E step that we have calculated something similar to the RHS of the equation. By inserting the results of the E Step, we get

$$i_{u,u'} = \frac{1}{\lambda_u}\sum_{v} h(u,u',v)$$

Calculating the E-Step and M-Step in an iterative manner until convergence, we derive the EM algorithm.

5.3 Complexity Analysis In Section 3, we show that the social and adoption graphs are sparse. That is, the number of edges in the graph is significantly smaller than the total number of possible edges, $|F| << N^2$ and $|E| << MN$. Since the graphs are sparse, our algorithm complexity should scale with respect to the number of edges instead of the number of vertices. We should also use sparse matrices to reduce the amount of memory required.

The efficiency of our learning algorithm can be easily seen from Equation 5.16 of the E Step and Equation 5.19 of the M Step. In the E Step, each user has to compute the latent variable $x_{v,u}$ for the number of items u has. The number of possible values $x_{v,u}$ can take depends on the number of social links u has. Based on this analysis, the upper bound complexity of E Step for each iteration in the EM algorithm is the product of number of users $|U| = N$, the maximum number of items a user has and the maximum number of friends a user has, $N\ .\ max(M_u)\ .\ max(N_u)$. The complexity of the M Step is similar to E Step so the overall complexity of each iteration is in $O(N\ .\ max(M_u)\ .\ max(N_u))$. We will empirically verify the running time and number of iterations for convergence in Section 6.5.

6 Experimental Evaluation

Our objective in the experiments is to evaluate the performance of our proposed methods in predicting missing adoption links. We first empirically decide the number of latent factors to use for the LDA decomposition. We then compare the performance of our proposed models with LDA for users with different self-dependency values and numbers of items. We illustrate the rate of convergence of the EM algorithm to show that our method is scalable and efficient. We show some case examples to illustrate how our proposed model works differently as compared with other methods. Finally, we look at the quality of the topics/latent factors learned.

6.1 Experimental Setup Data Set. For experiments, we extract data sets from the raw Epinions data set described in Section 3 and a separate LiveJournal data set. The LiveJournal data set was obtained by crawling livejournal.com to collect user profile pages. The initial crawled set corresponded to approximately 20% of active users in LiveJournal. The items in Epinions are products adopted by users while the items in LiveJournal are communities that the users join. Since our interest is in learning the correlation between social and adoption graphs, we prune the data set such that each user or item has a sufficient number of links in both graphs. Thus, we iteratively remove users with less than three incoming/outgoing links and items, and items with less than three users, until no such user/item can be found in the graphs. Table 3 shows the statistics of our Epinions and LiveJournal data sets.

Table 3: Statistics of our Data Subset

Name	#users	#items	#social links	#adoption links
Epinions	2,934	2,146	66,036	135,940
LiveJournal	3,773	21,463	209,832	216,586

The statistics in Table 3 shows that the Epinions data set and LiveJournal data set have different properties. The Epinions data set has a denser user-item adoption graph, while the LiveJournal data set has a denser user-user social graph. The two data sets will give a fair overview of how our models perform in predicting missing links under different scenarios.

Methods. In the experiments, we compare the following methods in terms of effectiveness.

- *LDA* represents the approach where a user relies only on her own latent factors.
- *Social Correlation* represents the approach using only social correlation (i.e., friends' latent factors).
- *Hybrid* represents the approach of using both a user's own latent factors as well as her friends'.

The formulations of these methods were given in Sections 4 and 5 respectively.

Metrics. We first hide 30% of the user item adoption links randomly in each data set to create a training set with the remaining links and a testing set with the missing links. Then for each method, we generate a ranking of adoption links for each user based on the probability values returned by the method. We then construct a Precision-Recall (PR) curve for each user, and measure the area under the PR curve (AUC). The performance of each method will be expressed relative to the *LDA* approach. The *AUC ratio* refers to the ratio of a method's AUC to *LDA*'s AUC. The higher the AUC ratio, the better a method performs relative to *LDA*.

6.2 Deciding the number of Latent Factors To decide the number of latent factors for factorizing, we measure the prediction performance of LDA while varying the number of latent factors. Then we measure the AUC of Precision and Recall (PR) curves for each latent factor. The choice of AUC PR over AUC ROC is because AUC PR gives a better measurement for skewed data [8]. The small AUC is an artifact of the extreme sparsity of the adoption matrices. With many more non-existent links than hidden links for testing, the task of predicting adoption links is naturally difficult.

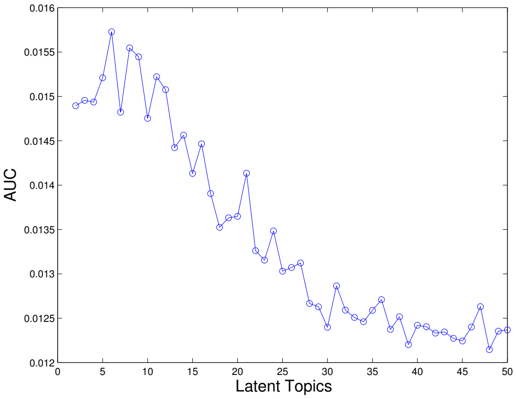

Figure 2: Epinions: Determine Number of Latent Topics

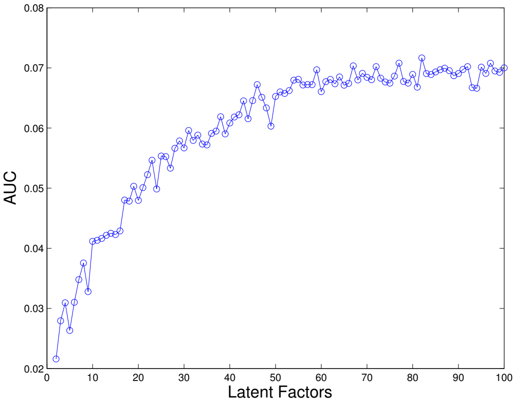

Figure 3: LiveJournal: Determine Number of Latent Topics

Figures 2 and 3 show the AUC with respect to the number of latent factors. As expected, the denser adoption graph in the Epinions data set only requires 6 latent factors and the sparser adoption graph in LiveJournal shows that the rate of AUC increase slows down around 40-50 latent factors. Although we can continue to increase the number of latent factors in LiveJournal to achieve better AUC performance, in consideration of memory usage and time, we will limit the number of latent topics to 45 for LiveJournal.

6.3 Self-Dependency Analysis Here, we showcase the merits of our proposed models by examining the AUC ratios for groups of users with varying self-dependency values. The diagonal values in I tell us how much each user depends on her own latent factors for items adoption. If a diagonal value $i_{u,u}$ is high, the corresponding user u is said to have a high self-dependency. Such a user is likely to adopt items based on her own latent factors. In contrast, a user with low self-dependency is likely to adopt items based on her friends' latent factors. We hypothesize that Social Correlation likely performs better than LDA for users with low self-dependency and Hybrid should do well on average for the different groups of users.

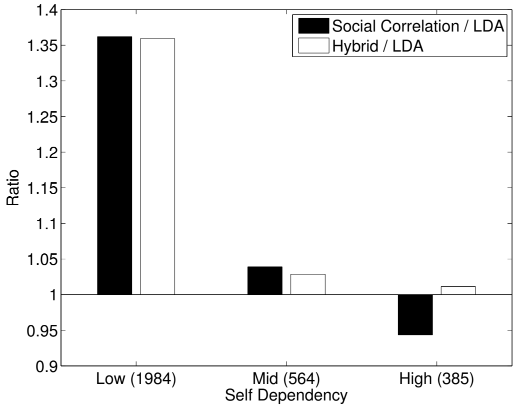

Figure 4: Epinions: AUC Ratio vs Self-Dependency

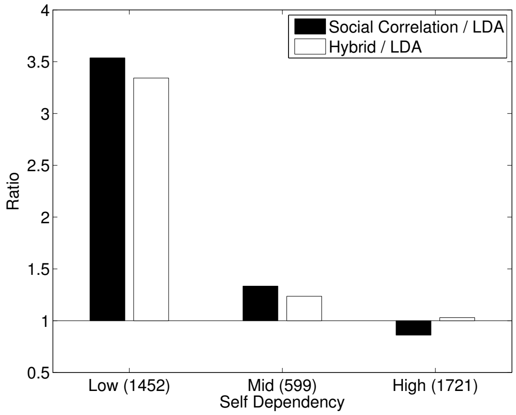

Figure 5: LiveJournal: AUC Ratio vs Self-Dependency

We bin the users into three groups of self-dependency values with low as $i_{u,u} \in [0, \frac{1}{3})$, mid as $i_{u,u} \in [\frac{1}{3}, \frac{2}{3}]$ and high as $i_{u,u} \in (\frac{2}{3}, 1]$. The bins interval are selected for them to be equal in size. We calculate for each user the AUC ratios $\frac{AUC\ Social\ Correlation}{AUC\ LDA}$ and $\frac{AUC\ Hybrid}{AUC\ LDA}$. Subsequently, we place each user in one of the low, mid, high self-dependency groups then calculate the mean of the ratios.

Figures 4 and 5 show the results of Epinions and LiveJournal for the mean ratios. In each figure, a higher bar indicates a better performance over the baseline method LDA. AUC ratio ≈ 1 means comparable performance with LDA, while higher ratios mean better performance over LDA. The number in parenthesis next to each self-dependency label indicates the number of users in that category.

In both figures, the results indicate that Social Correlation and Hybrid method work very well for users with low self-dependency values, showing significant improvement over LDA: $\geq 30\%$ for Epinions, and $\geq 200\%$ for LiveJournal. For users with mid self-dependency values, the improvements over LDA are more modest but still significant at about $3 - 20\%$. For users with high self-dependency, as expected, the results are very similar to LDA, with slight over-performance by Hybrid and slight under-performance by Social Correlation. These findings support our hypothesis that Social Correlation and Hybrid vastly improve upon LDA's performance, especially for users with low self-dependency values.

6.4 Number of Items Besides comparing with the self-dependency of each user, we also look at the AUC performance with respect to the number of items each user has. Figures 6 and 7 show the AUC ratio with respect to the log of the number of items (movies or communities) of the users. Users are organized into different groups based on the items that they have adopted. The black line parallel to the y-axis gives the median value for the number of items each user has. The figures show that Social Correlation helps to improve prediction for majority of the users in Epinions and approximately half of the users in LiveJournal. Hybrid helps to improve the prediction generally for most of the users in Epinions and LiveJournal. From these figures, we can also conclude that our methods (especially Hybrid) are very helpful for improving item adoption prediction for users with shorter adoption history (fewer items), while maintaining performance for users with longer adoption history.

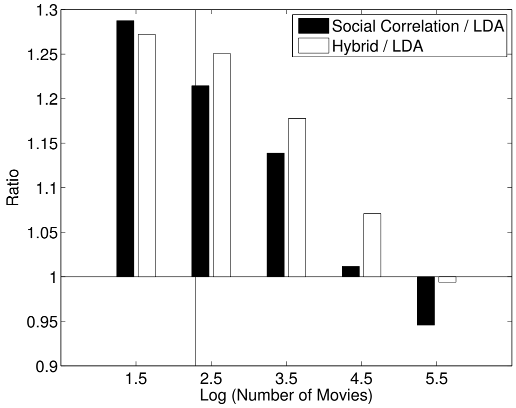

Figure 6: Epinions: AUC Ratio vs Log Movies

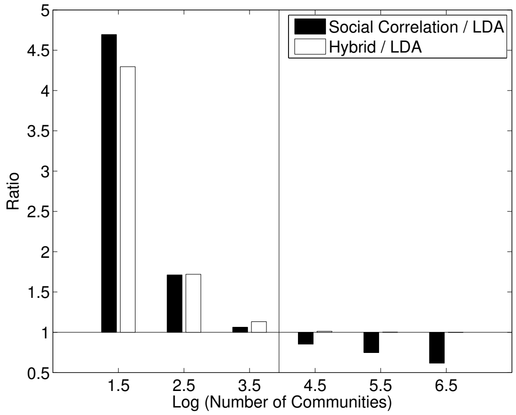

Figure 7: LiveJournal: AUC Ratio vs Log Communities

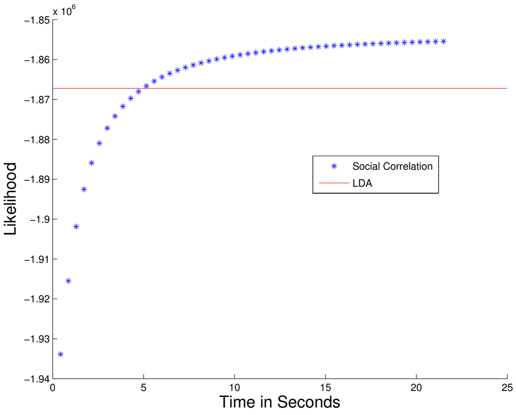

Figure 9: LiveJournal: Rate of MLE convergence

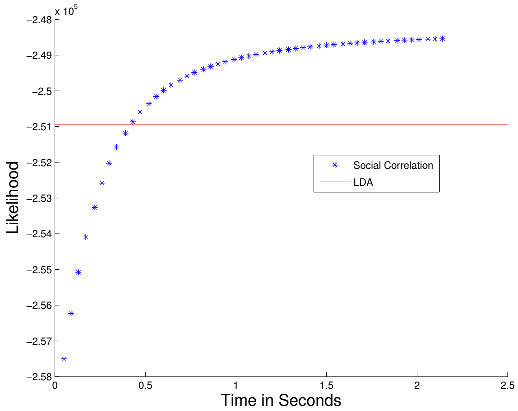

Figure 8: Epinions: Rate of MLE convergence

6.5 Convergence Rate We explained the complexity of the algorithm in Section 5.3. We now proceed to empirically verify that the EM algorithm for learning the social correlation matrix is able to converge by achieving a higher likelihood than LDA and is able to reach convergence relatively fast. We test our algorithm on a machine with Intel(R) Xeon(R) CPU X5460 @3.16GHz with 24 GB of memory.

Figures 8 and 9 show the likelihood with respect to time in seconds for Epinions and LiveJournal respectively. The likelihood is calculated using Equation 5.15. Since we have pre-computed LDA, the likelihood given by LDA is therefore a constant as shown by the red line in Figures 8 and 9. In the figures, each dot represents the likelihood of each iteration. As shown in the figures, it only takes a finite number of iterations for the likelihood of social correlation to exceed that of LDA. The time required for these iterations is also quite fast taking a couple of seconds to reach convergence. The figures show that the time taken for convergence on LiveJournal is ten times longer than Epinions. This observation reflects the correctness of the complexity we calculated earlier, $O(N\ .\ max(M_u)\ .\ max(N_u))$, where $max(M_u)$ represents the maximum number of items user u has. Since the number of items in LiveJournal is ten times larger than Epinions, then on average, each user in LiveJournal has more items than the users in Epinions.

6.6 Case Studies To illustrate how our proposed models work differently than other methods, we describe case studies involving two types of users: one with a low self-dependency (relying on friends for item adoption) and another with a high self-dependency (relying on own latent factors).

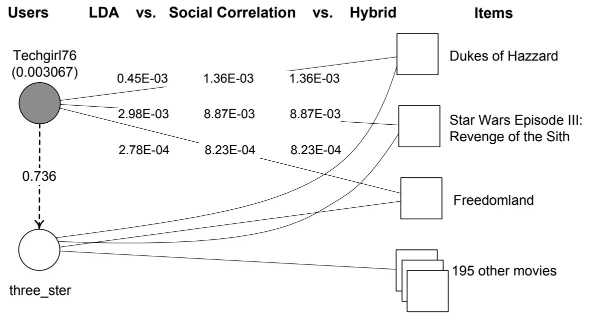

Figure 10: Epinions: Low Self-Dependency User

Low Self-Dependency. Figure 10 shows the profile of *techgirl76*, a user with low self-dependency as shown by the number in parentheses. *techgirl76* has adopted three items: Dukes of Hazzard, Star Wars III, and Freedomland. For each item, we show the probability of item adoption based on LDA versus Social Correlation and Hybrid. For all three adoption links, Social Correlation and Hybrid generate higher probability values, which suggest that *techgirl76*'s adoptions are highly motivated by friends' latent factors. This can be explained by the difficulty in learning *techgirl76*'s latent factors based on few items, as well as by *techgirl76*'s dependency on her friends. For instance, *techgirl76* has a very high dependency on another user *three_ster*, with $i_{u,u'} = 0.736$. *three_ster* has adopted the same three items as *techgirl76*, as well as another 195 items. The latent factors learned based on 198 items are likely to capture *three_ster*'s preferences well. Moreover, in addition to *three_ster*, there are also a couple of other friends who have adopted the three items. So it is likely that *techgirl76*'s adoptions are based on her friends' latent factors, rather than her own.

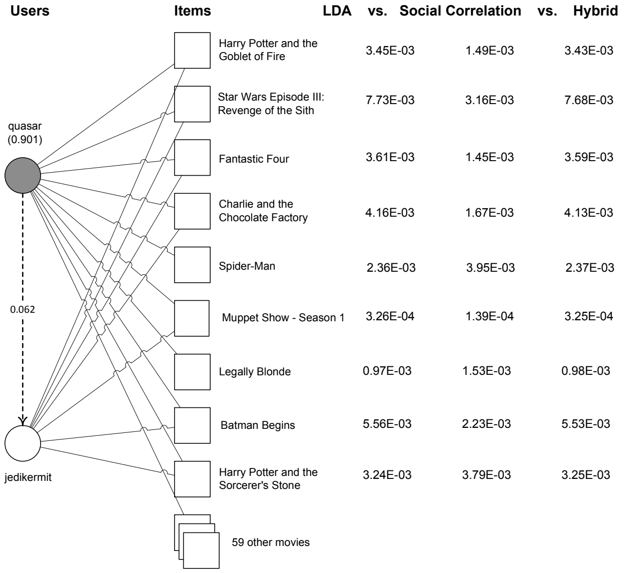

Figure 11: Epinions: High Self-Dependency User

High Self-Dependency. Figure 11 shows the profile of *quasar*, a user with high self-dependency as shown by the number in parentheses. In the figure, *quasar* shares nine items with her friends, one of whom is *jedikermit* shown in the figure. For most of the nine items, *quasar*'s probability of adoption based on LDA is higher than the probability based on Social Correlation. This suggests that *quasar* relies much more on her own latent factors than the latent factors of her friends. In addition, *quasar* has also adopted fifty nine other items that she does not share with any friend. Given the high number of items, LDA (and Hybrid) can learn the latent factors sufficiently well for *quasar*. Based on the latent factors, *quasar* actually likes most of the nine common items more than her friends do, which supports the case that *quasar* is a highly self-dependent user. This also explains why *quasar* has a low dependency on *jedikermit*, with $i_{u,u'} = 0.062$ even though she shares nine items with *jedikermit*.

6.7 Topic Analysis

Here, we evaluate the effectiveness of LDA in deriving the latent factors or topics. If LDA has learned the latent factors or topics well, each topic would correspond to a cluster of related items.

For ease of illustration, we only show three topics each for Epinions and LiveJournal. For each topic, we identify the top items with the highest latent factor values for that topic. Table 4 shows a sample of the top movie titles in each topic for the Epinions data set. The movies in each topic tend to be similar in terms of their genres. For instance, movies in *Topic E1* such as the Spider-Man and Lord of the Rings series are action movies. Movies in *Topic E2* are dramas such as Erin Brockovich and Fight Club. Movies in *Topic E3* seem to be comedies. Intuitively, these three topics also correspond to the three most popular genres in the data set: action, drama, and comedy.

Table 4: Example Top Movie Titles for Each Topic in Epinions

Topic E1	*Topic E2*	*Topic E3*
Spider-Man	Erin Brockovich	Shrek
Spider-Man 2	Fight Club	Charlie's Angels
Batman Begins	American Psycho	What Women Want
Lord of the Rings: The Two Towers	Magnolia	Meet the Parents
Lord of the Rings: The Return of the King	American Beauty	Miss Congeniality

Table 5 shows a sample of the top communities in each topic for the LiveJournal data set. The names of communities in LiveJournal draw from a wide variety of languages with Russian being a dominant language as seen by the prefix *ru_* in the communities name. *Topic L1* shows preference for East Asian culture. "jpop" is a synonym for Japanese Pop Music, "kpop" for Korean Pop Music, "jdramas" for Japanese Drama, "anime" and "manga" are terms for Japanese cartoons. *Topic L2* is of Information Technology subjects and *Topic L3* shows art and design.

Table 5: Example Top Communities for Each Topic in LiveJournal

Topic L1	*Topic L2*	*Topic L3*
free_manga	ru_webdev	ru_designer
anime_downloads	ru_linux	ru_photoshop
jdramas	ru_sysadmins	design_books
jpop_uploads	ru_software	ru_illustrators
kpop_uploads	ru_programming	ru_vector

7 Conclusion

In this paper, we address the problem of item adoption prediction based on both latent factors as well as social correlation. We incorporate a probabilistic social correlation matrix into a factorization approach based on LDA, and formulate two models: Social Correlation and Hybrid. To solve the models, we propose an efficient solution that scales with the number of observed links. Our models are based on several key ideas. In making item adoption choices, users are not motivated just by their own latent factors, but also by their friends'. The degree to which a user correlates to their friends' latent factors is not uniform, rather it differs from one user to another. Our experiments with Epinions and LiveJournal data sets show that the Social Correlation and Hybrid approaches outperform LDA.

There are several directions for future work. While in this work we have used LDA, we are also interested in investigating how the social correlation can also be used in conjunction with other factorization methods such as BPMF or SVD. Here we have focused very much on item adoption prediction based on social links, the reverse of the problem is equally interesting: whether we can predict social links based on item adoptions. Finally, the item adoption framework could potentially be extended to the rating prediction task.

8 Acknowledgement

This work is supported by Singapore's National Research Foundation's research grant, NRF2008IDM-IDM004-036.

References

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, 2008.

[2] L. Backstrom, D. Huttenlocher, and J. Kleinberg. Group formation in large social networks: Membership, growth, and evolution. In *KDD*, pages 44–54, 2006.

[3] R. Balasubramanyan and W. W. Cohen. Block-LDA: Jointly modeling entity-annotated text and entity-entity links. In *SDM*, 2011.

[4] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 2003.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[6] D. Cosley, D. Huttenlocher, J. Kleinberg, X. Lan, and S. Suri. Group formation in large social networks: Membership, growth, and evolution. In *ICWSM*, 2010.

[7] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *KDD*, pages 160–168, 2008.

[8] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *ICML*, 2006.

[9] 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.

[10] R. A. Fisher. On the interpretation of 2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922.

[11] T. La Fond and J. Neville. Randomization tests for distinguishing social influence and homophily effects. In *WWW*, pages 601–610, 2010.

[12] T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101, 2004.

[13] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

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

[15] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD*, pages 447–456, 2009.

[16] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In *CIKM*, 2010.

[17] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *SIGIR*, pages 203–210, 2009.

[18] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, pages 931–940, 2008.

[19] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2008.

[20] J. Neville and D. Jensen. Relational dependency networks. *J. Mach. Learn. Res.*, 8:653–692, 2007.

[21] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML*, pages 880–887, 2008.

[22] P. Singla and M. Richardson. Yes, there is a correlation: - from social networks to personal behavior on the web. In *WWW*, pages 655–664, 2008.

[23] J. Yang and J. Leskovec. Modeling information diffusion in implicit networks. In *ICDM*, 2010.