

## Singapore Management University Institutional Knowledge at Singapore Management University

---

Research Collection School Of Information Systems

School of Information Systems

---

9-2013

# Generative models for item adoptions using social correlation

Freddy Chong Tat CHUA

Singapore Management University, [freddy.chua.2009@smu.edu.sg](mailto:freddy.chua.2009@smu.edu.sg)

Hady Wirawan LAUW


Singapore Management University, [hadywlaw@smu.edu.sg](mailto:hadywlaw@smu.edu.sg)

Ee Peng LIM

Singapore Management University, [eplim@smu.edu.sg](mailto:eplim@smu.edu.sg)

**DOI:** <https://doi.org/10.1109/TKDE.2012.137>

Follow this and additional works at: [https://ink.library.smu.edu.sg/sis\\_research](https://ink.library.smu.edu.sg/sis_research)

 Part of the [Databases and Information Systems Commons](#), [Numerical Analysis and Scientific Computing Commons](#), and the [Social Media Commons](#)

---

### Citation

CHUA, Freddy Chong Tat; LAUW, Hady Wirawan; and LIM, Ee Peng. Generative models for item adoptions using social correlation. (2013). *IEEE Transactions on Knowledge and Data Engineering*. 25, (9), 2036-2048. Research Collection School Of Information Systems.

**Available at:** [https://ink.library.smu.edu.sg/sis\\_research/1550](https://ink.library.smu.edu.sg/sis_research/1550)

This Journal Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email [libIR@smu.edu.sg](mailto:libIR@smu.edu.sg).

# Generative Models for Item Adoptions Using Social Correlation

CHUA, Freddy Chong Tat; LAUW, Hady Wirawan; LIM, Ee Peng

## Abstract

Users face many choices on the Web when it comes to choosing which product to buy, which video to watch, etc. In making adoption decisions, users rely not only on their own preferences, but also on friends. We call the latter social correlation which may be caused by the selection and social influence effects. In this chapter, we focus on modeling social correlation on users item adoptions. Given a user-user social graph and an item-user adoption graph, our research seeks to answer the following questions: whether the items adopted by a user correlate to items adopted by her friends, and how to model item adoptions using social correlation. We propose a social correlation measure that considers the degree of correlation from every user to the users friends, in addition to a set of latent factors representing topics of interests of individual users. We develop two generative models, namely sequential and unified, and the corresponding parameter estimation approaches. From each model, we devise the social correlation only and hybrid methods for predicting missing adoption links. Experiments on LiveJournal and Epinions data sets show that our proposed models outperform the approach based on latent factors only (LDA).

## 1 Introduction

### 1.1 Motivation

Unprecedented progress and innovation provide consumers with 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 that consumers have a harder time making purchasing decisions, while sellers do not know what to sell and whom to sell it to.

Some merchants, such as Amazon and Netflix, have put in place 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 making adoption decisions, users 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 platforms such 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.

## 1.2 Objectives

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. A directed 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 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  $E$ .

Latent space approaches can model a user’s personal preferences [17]. One such model is Latent Dirichlet Allocation (LDA) [4], which learns a set of latent factors by reducing the adjacency matrix of the adoption graph into two sub components: one that reflects the importance of each latent factor to users, and another that does the same for items. However, this approach assumes that all items adopted by a user can be fully explained by the user’s and items’ latent factors.

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

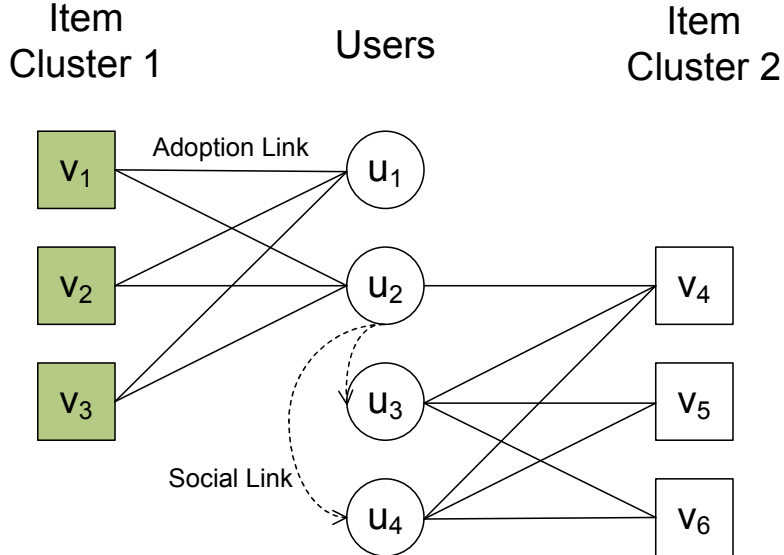


Figure 1: Example Scenario of Adoption (solid) and Social Links (dotted)

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 say that in the case of  $v_4$ ,  $u_2$  depends on the preferences of her friends  $u_3$  and  $u_4$ . We call this the *social correlation*.

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

### 1.3 Contributions

We make the following contributions in this paper:

1. We propose a Social Correlation Framework that incorporates the social correlation matrix  $C$  in the generation of user-item adoption links. Within this framework, we propose two generative models: *Sequential Generative Model* and *Unified Generative Model*. The Sequential Generative Model learns  $C$  in two sequential steps, first employing LDA to learn the parameters of the user and item latent factors, followed by learning  $C$  based on those parameters. The Unified Generative Model learns  $C$  simultaneously with the user and item latent factors in a prin-

ciplined, and unified way. The framework and two generative models are novel contributions over the previous state-of-the-art that relies only on user and item latent factors (e.g., LDA).

2. In our proposed generative models, the weights in the social correlation matrix are parameters to be learned. Hence, we do not rely on a social graph with pre-assigned link weights. This is essential because the weights are not always known. Even if some form of weights may be known (e.g., friendship strength), they may not accurately reflect the dependency weights among users for all domains of interest.
3. Through comprehensive experimentation on two real-life datasets (LiveJournal and Epinions), we establish that: (a) the proposed generative models under the social correlation framework outperform the approach that relies on latent factors alone, (b) the social correlation weights help to identify the users who will benefit most from social dependencies, and (c) the Unified Generative Model outperforms the Sequential Generative Model, which we attribute to the joint learning of parameters of the former generative model.

## 1.4 Organization

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 develop the Social Correlation Framework that incorporates social correlation in addition to the latent factors. In Sections 5 and 6, we describe two generative models under this framework: Sequential and Unified respectively, and show how their parameters can be learned efficiently. We then proceed to evaluate our methods in Section 7. Finally we conclude our paper in Section 8.

# 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 [20, 26] 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. Wen and Lin [33] show that combining different social media improves the social influence measure. McPherson et al. [25] 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 [30] 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.

The work by Liu et al. [22] sought to measure influence based on clearly observable following behaviors. Their technique is not applicable to our problem because of the following reasons. First, they only try to measure influence, but do not incorporate it to model item adoption. Second, they require stronger assumptions, whereby the directions of social edges are known, and where the influence direction is already known (e.g., Twitter users re-tweeted postings by others). Ours is a more generalized approach that allows any friend to be socially dependent on any friend. In such cases, the possible number of influencers can be very large and their method may not scale up.

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 [8, 2, 9]. 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 [24, 23]. 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. Instead of rating prediction, we model item adoption. Moreover, they assume the weighted values of social links are known (or assumed to be uniform). 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 [16, 6, 13, 7]. 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.

Leskovec et al. first analyzed the effect of social influence on viral marketing [21]. Then Yang and Leskovec addressed influence as a form of information diffusion [34] with temporal dynamics. Using a large collection of weblogs, Gruhl et al. studied the effects of information propagation through social networks [15]. However, their notion of influence requires the explicit adoption of item while we consider in terms of latent factors.

Snowsill et al. proposed a causal analysis of inferring social influence networks from data [31]. Their focus is on the social influence while we are motivated by the user item adoption problem. Cui et al. proposes a social influence matrix to suggest what items a user should share to maximize their individ-

ual influence in their own community [10]. Their matrix measures influence between users and items while ours measure between users and users.

## 2.2 Latent Space Approaches

The Bayesian Probabilistic Matrix Factorization (BPMF) is a popular model for low rank matrix approximation [27, 28] 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 [17].

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]. Wang and Blei proposed the combination of matrix factorization methods and topic models for the recommendation task but unlike our work, they do not consider the social network features.

## 2.3 Collaborative Filtering

Collaborative Filtering is a related field of research which also studies the relationships between users and items [18, 29, 32]. The main objective in collaborative filtering is to predict the rating a user will give to an item that has not been rated previously by the user. The rating will then determine whether a user will like the item and make a purchase eventually. Collaborative filtering has two important components: (a) determining the neighboring or users similar to the target user, and (b) personalizing the rating value based on the target user rating patterns. 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 [19]. Zhang et al. uses these approaches to demonstrate transfer learning among different collaborative filtering domains [35].

In our work here, our assumption is that users adopt items because of their friends' preferences or the amount of liking they have for the items. The social correlation which we compute between a pair of users indicates the strength of how much a user depends on her friend adoption patterns. Collaborative filtering also computes a similarity value between pairs of users but each pair may not necessarily have any social relationship. We distinguish our work by showing how the social relationships between users can be used to measure their social correlation which represents the strength of their friendship. Like BPMF, our proposed generative models can be further combined with neighborhood methods to derive a new collaborative filtering approach. This extension however is beyond the scope of this paper.



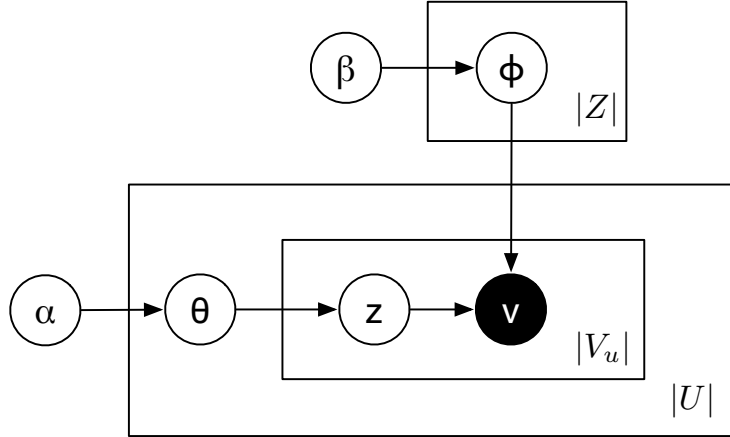


Figure 2: Latent Dirichlet Allocation in Plate Notation

## 2.4 LDA Generative Model

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. Figure 2 refers to the graphical notation of LDA. The generative process for LDA is as follows,

1. Each user  $u$  has a latent factor distribution  $\theta_u$  which indicates their preferences for a set of topics.  $\theta_u$  follows a symmetric Dirichlet distribution with hyper-parameters  $\alpha$ .

$$\theta_u \sim \text{Dirichlet}(\alpha)$$

2. For each item  $v$  that  $u$  adopts,  $u$  first chooses from a set of topics based on their topic preferences,

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

Then from the latent factors of items distribution  $\Phi$ ,  $u$  chooses the item  $v$  from as follows:

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

where  $\Phi$  follows a symmetric Dirichlet distribution with hyper-parameters  $\beta$ , as follows:

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

Solving for these parameters 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 [14].



Unlike BPMF which uses 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. Singla and Richardson [30] had also earlier established that correlations exist between friends on an online social messaging network. We investigate social correlation by performing hypothesis testing on two real world data sets obtained from LiveJournal, an online community site and Epinions, a product review site.

The social graph in LiveJournal consists of friendship links when a user indicates that another user is her friend [19]. These social links are directional and not necessarily reciprocal. An adoption link exists between a user and a community if the user has joined the community. 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. We only retain the users who have at least one social link and items who have at least one adoption. The size of the data sets is given in Table 1. In total, there are close to 16K users and 78K items for LiveJournal.

The social graph in Epinions consists of trust links formed when a user indicates her trust on another user. An adoption link exists between a user and a product if the user has written a review for the item for Epinions. We collected the Epinions data set by crawling the Epinions site, focusing only on the Videos & DVDs category. For both data sets, we only retain the users who have at least one social link and items who have at least one adoption. There are 13K users and 7K items for Epinions (see Table 1).

Table 1: Data Size

| Data set:                     | LiveJournal | Epinions |
|-------------------------------|-------------|----------|
| no. of users $ U $ :          | 16,376      | 12,895   |
| no. of items $ V $ :          | 78,129      | 6,543    |
| no. of adoption links $ E $ : | 63,160      | 83,763   |
| no. of social links $ F $ :   | 476,227     | 178,659  |

We perform hypothesis testing using the Fisher Exact Test [12]. 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  $H_1$ ,

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

|                 | No Common Adoption           | Has Common Adoption      | Total       |
|-----------------|------------------------------|--------------------------|-------------|
| No Social Link  | 131,281,395<br>(131,126,176) | 2,485,417<br>(2,640,636) | 133,766,812 |
| Has Social Link | 150,316<br>(305,535)         | 161,372<br>(6,153)       | 311,688     |
| Total           | 131,431,711                  | 2,646,789                | 134,078,500 |

Table 3: Epinions : Contingency Table For Pair of Users with Social and Adoption Links

|                 | No Common Adoption         | Has Common Adoption      | Total      |
|-----------------|----------------------------|--------------------------|------------|
| No Social Link  | 80,122,890<br>(80,103,462) | 2,874,403<br>(2,893,831) | 82,997,293 |
| Has Social Link | 112,575<br>(132,003)       | 24,197<br>(4,769)        | 136,772    |
| Total           | 80,235,465                 | 2,898,600                | 83,134,065 |

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 Tables 2 and 3. 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 161,372 is far greater than the expected 6,153 for LiveJournal. And the observed number of pairs with both common adoption and social link 24,197 is far greater than the expected 4,769 for Epinions.

Using Fisher Exact Test, we obtain a p-value  $< 2.2 \times 10^{-16}$  for both contingency tables which indicates that we can reject  $H_0$ , and conclude that the presence of social links is correlated with the presence of adoption links.

## 4 Social Correlation Measure

Our social correlation measure expresses the user-item adoptions  $E$  as a product of three components,  $\Phi$ ,  $\Theta$  and  $C^T$  as follows:

$$E \approx \Phi \cdot \Theta \cdot C^T \quad (1)$$

where  $\Phi$  represents the latent factors of items arranged in a  $|V| \times |Z|$  matrix with  $Z$  being the set of latent factors,  $\Theta$  represents the latent factors of users arranged in a  $|Z| \times |U|$  matrix, and  $C^T$  represents the tranpose of the  $|U| \times |U|$  social correlation matrix.

The social correlation measure requires us to determine all user-item adoptions and the three matrix components. If some elements of  $E$  can be observed, we can use them to learn the matrix components by minimizing the error  $|E - \Phi \cdot \Theta \cdot C^T|$ . This is akin to maximizing 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.

## 4.1 Social Correlation Matrix

The  $|U| \times |U|$  *social correlation* matrix  $C$  tells us how likely a user will adopt an item based on the latent factors of other users. Each element  $c_{u,u'}$  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  $u'$ .  $c_{u,u}$  is the **self-dependency** of user  $u$ , or the likelihood that  $u$  relies on her own latent factors. Each user has a set of social correlation values where each social correlation value defines the correlation between the user and one of her neighbor. This social correlation tells us how likely the user will follow the actions of her neighbor. The self-dependency value, is the social correlation value between the user and the user herself (because the user can be a neighbor of herself). A high self-dependency value indicates that the user is very independent in making adoption decisions and will not follow other users easily. A low self-dependency value indicates that the user depends on her friends for making adoption decisions.

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

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

## 4.2 Probabilistic Formulations

We would like to illustrate the formulation of our models using probabilistic explanations. Given a user  $u$ , we would like to know the probability that she will adopt the item  $v$ , given the users latent factors  $\Theta$  and the latent factors of items  $\Phi$ .

Suppose now that we have the edges of the social graph  $F$  and the latent factors of all users in  $U$  including herself, we hypothesize that the user  $u$  adopts items based on the latent factor preferences of her friends  $F_u^v$  and the

user herself. We may restate the equation as follows,

$$\begin{aligned} P(e_{v,u}|\Theta, \Phi, F) &= \sum_{x \in F_u} P(e_{v,x}, f_{u,x}|\Theta, \Phi, F) \\ &= \sum_{x \in F_u} P(e_{v,x}|\Theta, \Phi)P(f_{u,x}|F) \end{aligned} \quad (2)$$

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

Equation 2 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  $x$ . The reason is besides knowing the probability that  $u$  indicates  $x$  as a friend in the social graph  $P(f_{u,x}|F)$  and the probability that  $x$  adopts item  $v$  in the adoption graph  $P(e_{v,x}|\Theta, \Phi)$ , we need a weighted component that tells us the probability that  $u$  depends on  $x$  in the adoption graph  $P(x_{v,u} = x|C, F)$  (to be defined shortly). This component is the social correlation that we want to determine.

Hence, our proposed latent space model is to introduce the latent variable  $x_{v,u}$  which tells us which  $x$  that  $u$  depends on to adopt  $v$ , and the social correlation  $C$  where its elements  $c_{u,x}$  gives us the probability that  $u$  follows the latent factors of  $x$ . The special case is  $x = u$  which tells us the self-dependency of  $u$ . The higher  $c_{u,u}$  is, 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  $C$  is given by:

$$\begin{aligned} P(e_{v,u}|\Theta, \Phi, F, C) &= \sum_{x \in F_u} P(e_{v,x}, x_{v,u} = x|\Theta, \Phi, F, C) \\ &= \sum_{x \in F_u} P(e_{v,x}|\Theta, \Phi)P(x_{v,u} = x|C, F) \end{aligned}$$

where  $F$  the social network is always available. The information to be learnt are  $\Theta$ ,  $\Phi$  and  $C$ .

### 4.3 Prediction Models

Once the social correlation matrix  $C$  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  $\Phi \cdot \Theta \cdot C^T$  (see Equation 1) based on the learned  $C$ , taking into account only the non-diagonal values of  $C$ , i.e., setting  $c_{u,u} = 0, \forall u \in U$ .
- *Hybrid* represents the approach of combining Social Correlation and LDA, by computing  $\Phi \cdot \Theta \cdot C^T$  with the original learned  $C$  (with diagonal values retained).

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

## 5 Sequential Generative Model

The Sequential Generative Model assumes that the values  $e_{v,u}$  is adequately estimated by the LDA. This assumption is reflected by the shaded  $\theta$  and  $\phi$  variables in the graphical model as shown in Figure 3. We also assume the existence of a social network as reflected by the shaded  $f$  variables.

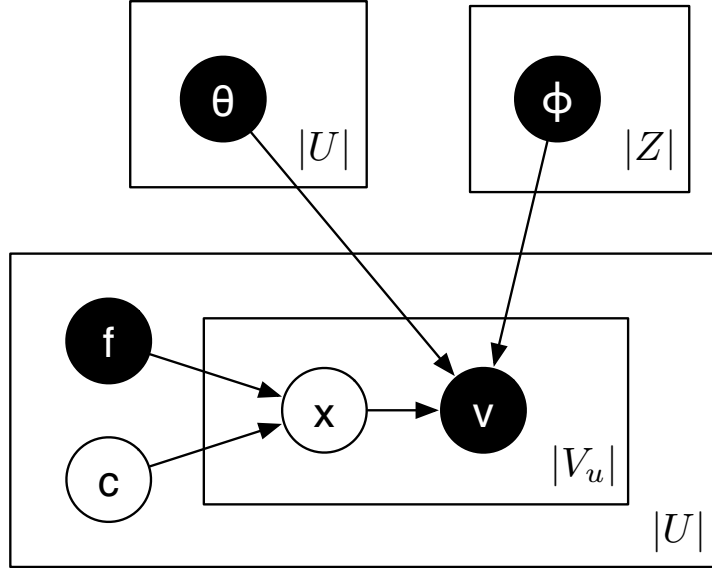


Figure 3: Sequential Generative Model for Static Social Correlation

$C$  can be obtained in several ways. The naive way is to calculate  $C$  by multiplying  $E$  with the inverse of  $\Phi \cdot \Theta$ , i.e.  $C = (\Phi \cdot \Theta)^{-1} \cdot E$ . This naive way will not work for several reasons.

1.  $C$  may over-fit leading to poor results in link prediction. The obtained  $\Phi \cdot \Theta \cdot C^T$  will be as sparse as  $E$ , and thus the factorization does not help in link prediction.
2.  $C$  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.
3.  $C$  may have non-zero values even if the users are not connected by social links.

Instead of this naive way, we devise a generative model called *Sequential Generative Model*, with the following generative process,

1. For a given user  $u$ ,  $u$  chooses a friend  $x$  from her set of friends  $F_u$  and her social correlation with that friend  $c_{u,x}$  for adopting the item  $v$ .

$$P(x_{v,u} = x | C_u, F_u) = c_{u,x}$$

2. Given the known probability of user  $x$  adopting item  $v$ ,  $u$  adopts  $v$  based on how likely  $x$  adopts item  $v$ ,

$$P(e_{v,x}|\Theta, \Phi) = \sum_z \theta_{x,z} \cdot \phi_{z,v}$$

where above equation has parameters  $\Theta$  and  $\Phi$  computed by LDA.

The probability of user  $u$  adopting item  $v$  is therefore:

$$\begin{aligned} P(e_{v,u}|\Theta, \Phi, F, C) &= \sum_{x \in F_u} P(e_{v,x}|\Theta, \Phi)P(x_{v,u} = x|C_u, F_u) \\ &= \sum_{x \in F_u} e'_{v,x} c_{u,x} \end{aligned}$$

and  $e'_{v,x}$  is the  $(v, x)$  element of  $\Phi \cdot \Theta$ .

To learn the social correlation values, we maximize the log likelihood of  $e_{v,u}, \forall u \in U, \forall v \in V_u$ , using the Expectation Maximization (EM) algorithm [11],

$$\begin{aligned} \log P(E|\Theta, \Phi, F, C) &= \sum_{u,v} \log P(e_{v,u}|\Theta, \Phi, F, C) \\ &= \sum_{u,v} \log \sum_x e'_{v,x} c_{u,x} \end{aligned}$$

where  $\sum_{u,v}$  is short for  $\sum_{u \in U} \sum_{v \in V_u}$ .  $U$  represents the set of users in our data and  $V_u$  represents the set of items  $V_u$  adopted by user  $u$ .

## 5.1 Expectation Maximization Algorithm

We first show the E Step. The E Step of the EM algorithm infers the latent variables using initial values of  $C$ ,

$$\begin{aligned} P(x_{v,u} = x|e_{v,u}, \Theta, \Phi, F, C) &= \frac{P(e_{v,x}|\Theta, \Phi)P(x|C_u, F_u)}{\sum_{x' \in F_u} P(e_{v,x'}|\Theta, \Phi)P(x'|C_u, F_u)} \\ &= \frac{e'_{v,x} c_{u,x}}{\sum_{x' \in F_u} e'_{v,x'} c_{u,x'}} \\ &= h(u, x, v) \end{aligned} \tag{3}$$

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

$$\sum_{x \in F_u} c_{u,x} = 1, \quad \forall x \in U$$

Now for the M step, we aim to maximize the log likelihood with respect to the unknown social correlation  $C$ , subject to the above constraints. In order to include the constraints as part of the objective function, we introduce the

Lagrange multipliers  $\lambda_u$  [5] and proceed to solve the following using differentiation,

$$\begin{aligned}
\frac{d}{d c_{u,x}} \left[ \sum_{v \in V_u} \sum_{u \in U} \log \left( \sum_{x \in F_u} e'_{v,x} c_{u,x} \right) - \lambda_u \left( \sum_{x \in F_u} c_{u,x} - 1 \right) \right] &= 0 \\
\sum_{v \in V_u} \frac{e'_{v,x}}{\sum_{x' \in F_u} e'_{v,x'} c_{u,x'}} - \lambda_u &= 0 \\
\lambda_u &= \sum_{v \in V_u} \frac{e'_{v,x}}{\sum_{x' \in F_u} e'_{v,x'} c_{u,x'}} \\
\lambda_u c_{u,x} &= \sum_{v \in V_u} \frac{e'_{v,x} c_{u,x}}{\sum_{x' \in F_u} e'_{v,x'} c_{u,x'}} \\
c_{u,x} &= \frac{1}{\lambda_u} \sum_{v \in V_u} \frac{e'_{v,x} c_{u,x}}{\sum_{x' \in F_u} e'_{v,x'} c_{u,x'}} \tag{4}
\end{aligned}$$

Recall in our E step that we have calculated something similar to the RHS of Equation 4. By inserting the results of Equation 3 from the E Step, we get

$$c_{u,x} = \frac{1}{\lambda_u} \sum_{v \in V_u} h(u, x, v)$$

where  $\lambda_u$  can be seen as a normalizing constant. Calculating the E Step and M Step in an iterative manner until convergence, we derive the EM algorithm.

## 5.2 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,  $|F| \ll |U|^2$  and  $|E| \ll |V| \cdot |U|$ . 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 3 of the E Step and Equation 4 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 complexity of the Sequential Estimation is therefore given by,  $O(|U| \cdot \text{avg}(|V_u|) \cdot \text{avg}(|F_u|))$ . Expressing in terms of number of edges,

$$\begin{aligned}
O(|U| \cdot \text{avg}(|V_u|) \cdot \text{avg}(|F_u|)) &= O \left( \frac{|U| \cdot \text{avg}(|V_u|) \cdot |U| \cdot \text{avg}(|F_u|)}{|U|} \right) \\
&= O \left( \frac{|E| \cdot |F|}{|U|} \right)
\end{aligned}$$

Complexity for each iteration of our EM algorithm is given by  $O \left( \frac{|E| \cdot |F|}{|U|} \right)$ . We will empirically verify the running time and number of iterations for convergence in Section 7.5.



## 6 Unified Generative Model

The Sequential Model performs the derivation of latent factors and social correlation variables separately for simplicity. Following the model semantics, the social correlation parameters requires knowledge of the latent variables  $x_{v,u}$  which can only be estimated accurately given the latent variables  $z_{v,u}$ . However, the latent variables  $z_{v,u}$  also depend on the value of  $x_{v,u}$ . This circular dependency complicates the learning of the latent variables and their respective parameters:  $\Theta$ ,  $\Phi$  and  $C$ . The sequential approach we took in Section 5, gives us a simple approach to estimating  $x_{v,u}$  and additional assurance that once the latent variables  $z_{v,u}$  have been adequately estimated, estimation of  $x_{v,u}$  will lead to a better overall performance of the model. In this section, we proposed a unified estimation for the parameters of  $\Phi$ ,  $\Theta$  and  $C$ .

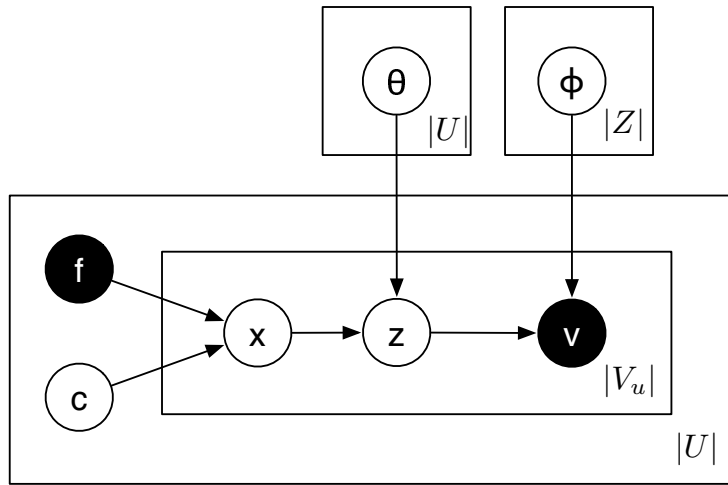


Figure 4: Unified Generative Model for Static Social Correlation

Figure 4 shows the plate notation of our graphical model. In this paper, we provide a unified way of learning the latent variables  $x$  and  $z$  using the Expectation Maximization (EM) approach for learning two sets of latent variables. We describe the generative process as follows,

1. For a given user  $u$ ,  $u$  chooses a friend  $x$  from her set of friends  $F_u$  and her social correlation with that friend  $c_{u,x}$  for adopting the item  $v$ .

$$P(x_{v,u} = x | C_u, F_u) = c_{u,x}$$

2. From the chosen friend  $x$ , who may be  $u$  herself,  $u$  chooses a latent factor  $z_{v,u}$  based on the latent preferences of the chosen friend  $\theta_x$ .

$$P(z_{v,u} = z | x_{v,u} = x, \Theta) = \theta_{x,z}$$

3. Finally, given the latent factor  $z_{v,u}$  and the latent factor items  $\phi_z$ ,  $u$  chooses an item  $v$  to adopt.

$$P(e_{v,u} | z_{v,u} = z, \Phi) = \phi_{z,v}$$

## 6.1 Parameter Estimation

Given a user-item matrix  $E$ , a social network  $F$ , a set of users  $U$ , a set of items  $V$ , let  $u \in U$  denote a user,  $v \in V$  denote an item, the element  $e_{v,u} = 1$  of matrix  $E$  denote that  $u$  adopts item  $v$ . Suppose we have a user to user correlation matrix  $C$ , where  $c_{u,x} > 0$  if  $u, x \in U$  and  $u$  is friends with  $x$ . Details of the derivation is given in Appendix A.

The E Steps are

$$f(u, v, z) = \frac{\phi_{z,v} \theta_{u,z} c_{u,u}}{\sum_{z' \in Z} \sum_{x' \in F_u} \phi_{z',v} \theta_{x',z'} c_{u,x'}} \quad (5)$$

$$g(u, v, z) = \frac{\sum_{x \in F_u} \phi_{z,v} \theta_{x,z} c_{u,x}}{\sum_{z' \in Z} \sum_{x' \in F_u} \phi_{z',v} \theta_{x',z'} c_{u,x'}} \quad (6)$$

$$h(u, v, x) = \frac{\sum_{z \in Z} \phi_{z,v} \theta_{x,z} c_{u,x}}{\sum_{z' \in Z} \sum_{x' \in F_u} \phi_{z',v} \theta_{x',z'} c_{u,x'}} \quad (7)$$

The M Steps are,

$$\theta_{u,z} = \frac{1}{\gamma_u} \sum_{v \in V_u} f(u, v, z)$$

$$\phi_{z,v} = \frac{1}{\delta_z} \sum_{u \in U} g(u, v, z)$$

$$c_{u,x} = \frac{1}{\lambda_u} \sum_{v \in V_u} h(u, v, x)$$

## 6.2 Complexity Analysis

As mentioned in Section 5.2, it suffices to analyze the complexity of the E Step, so we shall focus on the E Step for the Unified Estimation method. The E Steps of Unified Estimation depends on Equations 5, 6 and 7. For each user  $u$ , Equations 5, 6 and 7 requires  $O(|Z| \cdot |V_u| \cdot |F_u|)$ . So the complexity for all users is given by  $O(|U| \cdot |Z| \cdot \text{avg}(|V_u|) \cdot \text{avg}(|F_u|))$ . Following the previous analysis on the complexity, The complexity is given by,

$$O\left(\frac{|Z| \cdot |E| \cdot |F|}{|U|}\right)$$

# 7 Experimental Evaluation

## 7.1 Experimental Setup

**Data Set:** For experiments, we extracted data sets from the LiveJournal data set and Epinions data set described in Section 3. The items in LiveJournal are communities that the users join, while the items in Epinions are products reviewed by users. Also recall that Epinions has user-user trust links while LiveJournal has user-user friendship links.

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. We need such a minimum threshold so that when we divide the data sets into training and testing sets, each user and item will at least have some links to hold out for testing. Table 4 shows the statistics of our LiveJournal and Epinions data sets. The size of our dataset here is smaller than the size as shown in Table 1 due to the pruning steps as mentioned above. It is necessary for the pruning because it will be difficult to learn the latent factors of users with fewer than three items.

Table 4: Statistics of our Data Subset

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

The statistics in Table 4 shows that the LiveJournal data set and Epinions data set have different properties. The LiveJournal data set has a denser user-user social graph, while the Epinions data set has a denser user-item adoption 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.

- *Random* represents the approach where we randomly predict the items that a user will adopt. This is our baseline method for obtaining a performance ratio.
- *LDA* represents the approach where a user relies only on her own latent factors and also latent factor of items.
- *Sequential Social* represents the approach using only social correlation (i.e., friends’ latent factors), and parameters estimated using the Sequential Model Method.
- *Sequential Hybrid* represents the approach of using both a user’s own latent factors as well as her friends’, and parameters estimated using the Sequential Model Method.
- *Unified Social* represents the approach using only social correlation (i.e., friends’ latent factors), and parameters estimated using the Unified Model Method.
- *Unified Hybrid* represents the approach of using both a user’s own latent factors as well as her friends’, and parameters estimated using the Unified Model Method.

At times, we may need to refer to two methods as a group. In those cases, we use a short form of *Sequential* to refer to both the *Sequential Social* and *Sequential Hybrid*. Similarly for *Unified*. On the other hand, *Social* is a short

form to refer to both *Sequential Social* and *Unified Social*. Similarly for *Hybrid*. The formulations of these methods were given in Section 2.4 (*LDA*), Section 5 (*Sequential*), and Section 6 (*Unified*) 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 adoption 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 *AUC ratio* refers to the ratio of a method’s AUC to *Random*’s AUC. The higher the AUC ratio, the better a method performs relative to *Random*. The performance of each method is therefore defined to be the average of AUC or AUC ratio over all users.

## 7.2 Number of Latent Factors

To decide the number of latent factors for factorizing, we measure the prediction performance of *LDA*, *Sequential Social*, *Sequential Hybrid*, *Unified Social* and *Unified Hybrid* using their aggregated AUC results of all users, while varying the number of latent factors.

Figures 5 and 6 show the AUC with respect to the number of latent factors. *Unified Hybrid* outperforms *Sequential Hybrid* and *LDA* for all factors. *Unified Social* outperforms *Sequential Social* for all factors.

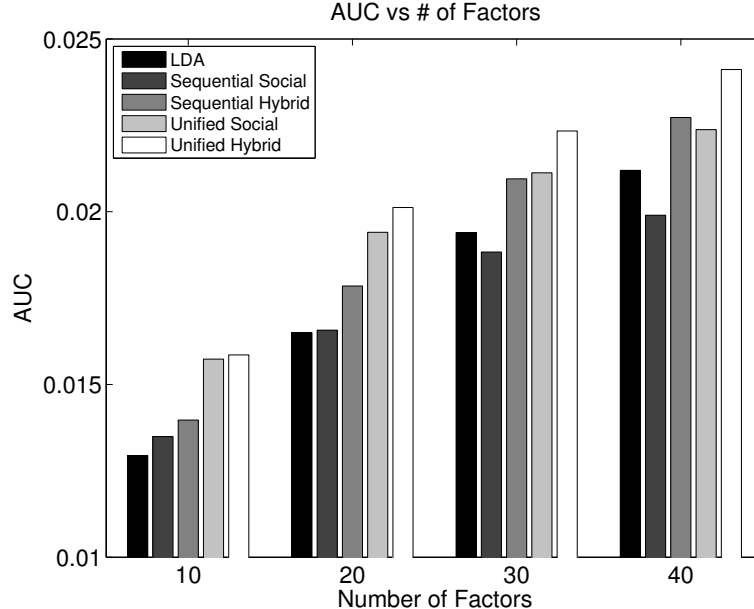


Figure 5: LiveJournal: AUC vs Number of Factors

Since our performance is consistent across all latent factors, for the rest of the experiments in this section, we pick 40 latent factors for LiveJournal and 10 latent factors for Epinions because they are manageable numbers for computation and are reasonable numbers for the size of the data sets.

Appendix B shows the list of top ranked items for a subset of the topics.

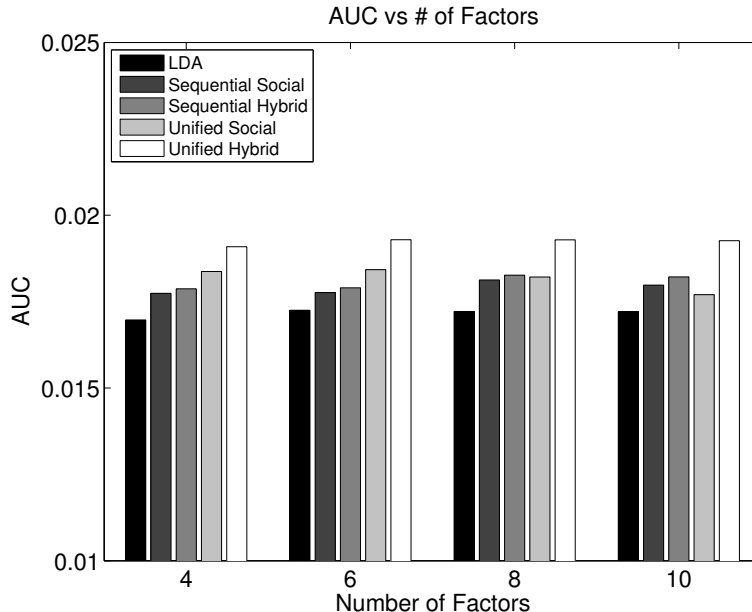


Figure 6: Epinions: AUC vs Number of Factors

The items in these topics give us a qualitative view of whether the chosen number of topics is appropriate.

### 7.3 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. Given that we have the *Sequential Model* and *Unified Model* of deriving the self-dependencies, we only compare for *LDA* vs *Sequential Social* vs *Sequential Hybrid* and *LDA* vs *Unified Social* vs *Unified Hybrid*. The diagonal values in  $C$  tell us how much each user depends on her own latent factors for items adoption. If a diagonal value  $c_{u,u}$  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* likely performs better than *LDA* for users with low self-dependency and *Hybrid* should do well on average for the different groups of users.

We bin the users into three groups of self-dependency with *low* as  $c_{u,u} \in [0, \frac{1}{3})$ , *mid* as  $c_{u,u} \in [\frac{1}{3}, \frac{2}{3}]$  and *high* as  $c_{u,u} \in (\frac{2}{3}, 1]$ . The bins interval are selected for them to be equal in width. We calculate for each user the AUC ratios  $\frac{AUC_{Social}}{AUC_{Random}}$  and  $\frac{AUC_{Hybrid}}{AUC_{Random}}$ . Subsequently, we place each user in one of the *low*, *mid*, *high* self-dependency groups then prune away the top 95 percentile and bottom 5 percentile to calculate the trimmed mean of the ratios.

Figures 7 and 8 show the results of LiveJournal and Epinions for the mean ratios using the *Sequential Model*. In each figure, a higher bar indicates a better performance over the baseline method *Random*. AUC ratio  $\approx 1$  means comparable performance with *Random*, while higher ratios mean better performance over *Random*. The number in parenthesis next to each self-dependency label indicates the number of users in that category.

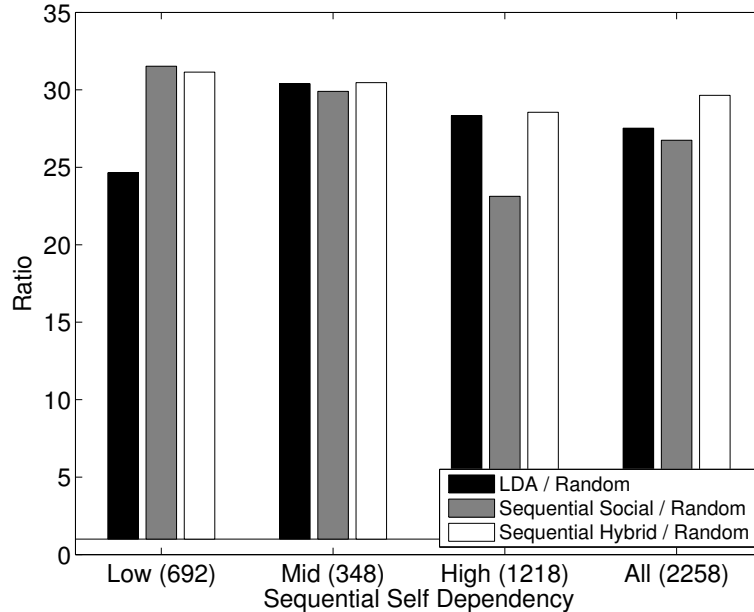


Figure 7: LiveJournal: Sequential Model AUC Ratio vs Self-Dependency

In both figures, the results indicate that *Social* and *Hybrid* methods work very well for users with low self-dependency values, showing significant improvement over *LDA*. For users with mid self-dependency values, the improvements over *LDA* are more modest. For users with high self-dependency, as expected, the results of *Hybrid* are very similar to *LDA*, with slight over-performance by *Hybrid* and slight under-performance by *Social*. These findings support our hypothesis that *Social* and *Hybrid* vastly improve upon *LDA*'s performance, especially for users with low self-dependency values. The performance of *Hybrid* over *Social* increases as the self-dependency increases. This suggests that friend's preferences matters less to users of high self-dependency.

Figures 9 and 10 show the results of LiveJournal and Epinions for the mean ratios using the *Unified Model*. In both figures, the results indicate that our models work well for users with low self-dependency values. As self dependency increases, the edge unified has over *LDA* decreases as expected. These findings are also similar to that of the *Sequential Model*. Consistent with the *Sequential Model* results, the *Hybrid* performance increases over *Social* as self-dependency increases.

We are not able to compare side-by-side the performance of *Sequential Model* and *Unified Models* with respect to the self-dependency values because the self-dependency values are specific only to each method. In the following section, we will compare the performance of *Sequential* and *Unified* with respect to the number of items each user has. Please also refer to Appendix C for further analysis on the self-dependency values.

## 7.4 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 11 and 12 show the AUC ratio with respect to the log of the number of items

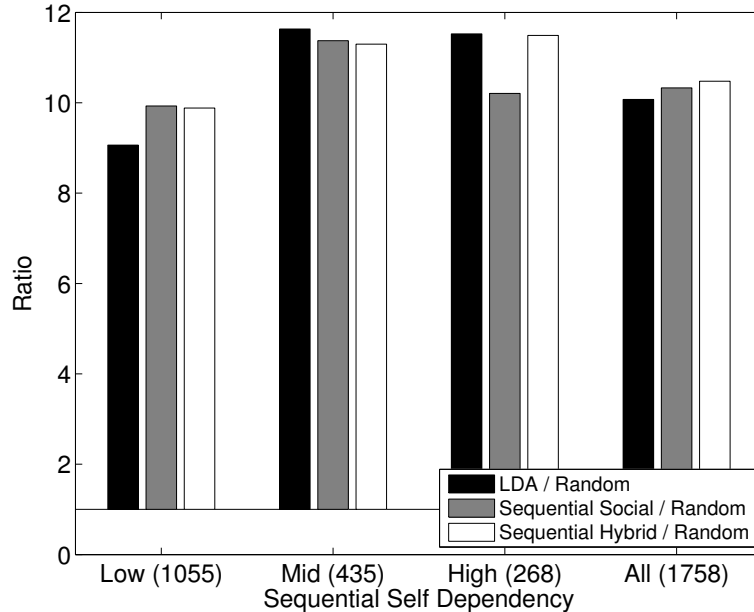


Figure 8: Epinions: Sequential Model AUC Ratio vs Self-Dependency

(communities or movies) of the users. Users are organized into different groups based on the number of items that they have adopted. The vertical-line parallel to the y-axis gives the median value for the number of items each user has. As shown in Figure 11 for LiveJournal, Social outperforms LDA for approximately half of the users. For Figure 12 for Epinions, Social outperforms LDA in the first three bins (beyond the median), effectively improving prediction for more than half of the users. The figures show that *Social* improves prediction for a majority of the users in Epinions and approximately half of the users in LiveJournal. *Hybrid* improves the prediction accuracy for even more users in LiveJournal and Epinions. 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.

Figures 11 and 12 show that the performance of our models with respect to *Random* decrease when number of items adopted by the user increases. This decrease in relative performance is because *Random* has better performance when there are more items to predict. Theoretical estimate of *Random*'s AUC with respect to the number of items can be found in D.

## 7.5 Convergence Rate

We explained the complexity of the algorithm in Section 5.2 and Section 6.2. 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 13 and 14 show the likelihood with respect to number of iterations for LiveJournal and Epinions respectively. Since we have pre-computed *LDA*



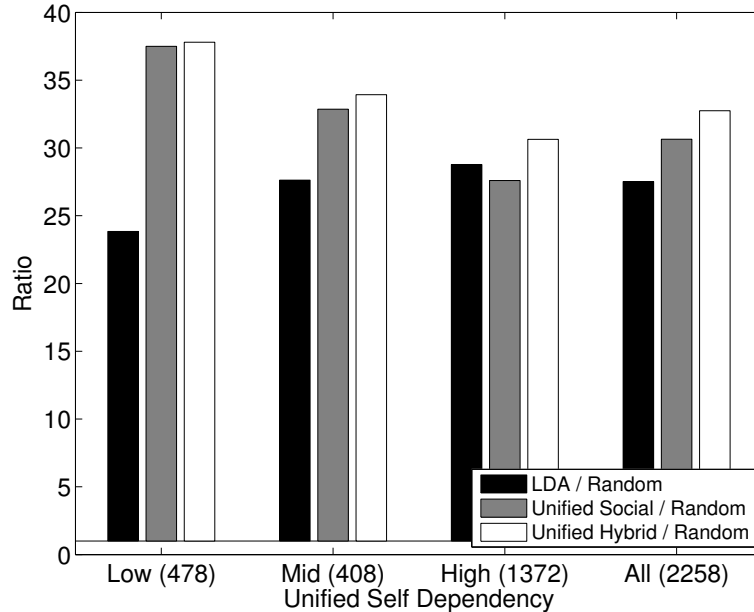


Figure 9: LiveJournal: Unified Model AUC Ratio vs Self-Dependency

for the *Sequential Model*, the likelihood given by *LDA* is therefore a constant as shown by the red line in Figures 13 and 14. In the figures, each dot represents each iteration. As shown in the figures, it only takes a small number of iterations for the likelihood of *Sequential Model* and *Unified Model* to exceed that of *LDA*. The time required for these iterations is also quite fast taking a couple of seconds to reach convergence.

For LiveJournal, *LDA* took 547 seconds, each iteration of *Sequential Model* 0.315 seconds and each iteration of Unified Model took 365 seconds. For Epinions, it took about 6.1 seconds to run *LDA*, each iteration of *Sequential Model* took 0.0313 seconds, each iteration of Unified Model took 3.49 seconds. Hence, the *Sequential Model* takes less time to be learnt compared with the Unified Model (assuming that each model requires at least 5 iterations). The *LDA* model requires the least amount of time.

## 7.6 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). To avoid repetition of analysis and space constraints, we only show the case studies for the *Unified Social* and *Unified Hybrid* for the LiveJournal data set.

**Low Self-Dependency.** Figure 15 shows the profile of *starkoff*, a user with low self-dependency ( $c_{u,u} = 0.19$ ) as shown by the number in parentheses. *starkoff* has adopted twenty four items, in which eight of these items are also adopted by *starkoff*'s friends, *uletelisamolety* and *ruslash*. For each prediction method, we show the items' ranks based on adoption probabilities generated by the method. In other words, the higher the probability of adopting the item, the smaller the number (rank). Since these items are the true adoptions by the

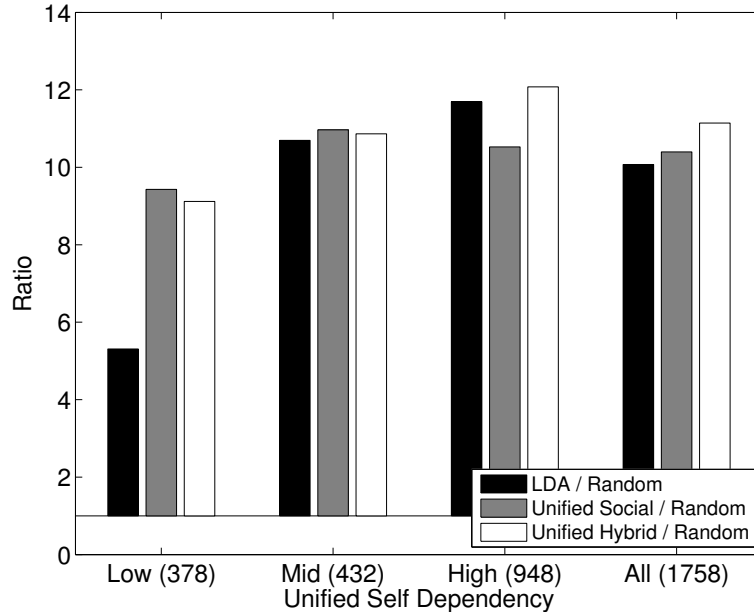


Figure 10: Epinions: Unified Model AUC Ratio vs Self-Dependency

user, a smaller rank implies a stronger result. As shown by the ranks, seven out of eight items gives a better or equal rank when we apply the probabilities given by *Unified Social* and *Unified Hybrid*. This suggests that *starkoff*'s adoptions are highly motivated by friends' latent factors and the social correlation for a user friends is important to suggest items of adoption for low self-dependency users.

**High Self-Dependency.** Figure 16 shows the profile of *\_prmarker*, a user with high self-dependency ( $c_{u,u} = 0.953$ ). *\_prmarker* adopts fifty nine items where four of these items are also adopted by her friends. The ranks of these four items show that we should use either *LDA* or *Unified Hybrid* to predict for their adoption. *Hybrid* is better than *Social* for these four items while *LDA* is better than *Social* and *Hybrid* for three out of four items. In addition to these four shared items, we also show five other items that *\_prmarker* does not share with her friends. In these five items, the ranks indicate that *Hybrid* is better than *LDA* which in turn is better than *Social*. This suggests that the social correlation is less important for high self-dependency users.

## 8 Summary

In this chapter, we address the problem of modeling item adoptions based on social correlation. We propose a social correlation measure that incorporates a probabilistic social correlation matrix into a latent space approach. Our social correlation is 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. We design two generative models: *Sequential Generative Model* that learns the social correlation matrix and latent factors in two steps, and *Unified Generative*

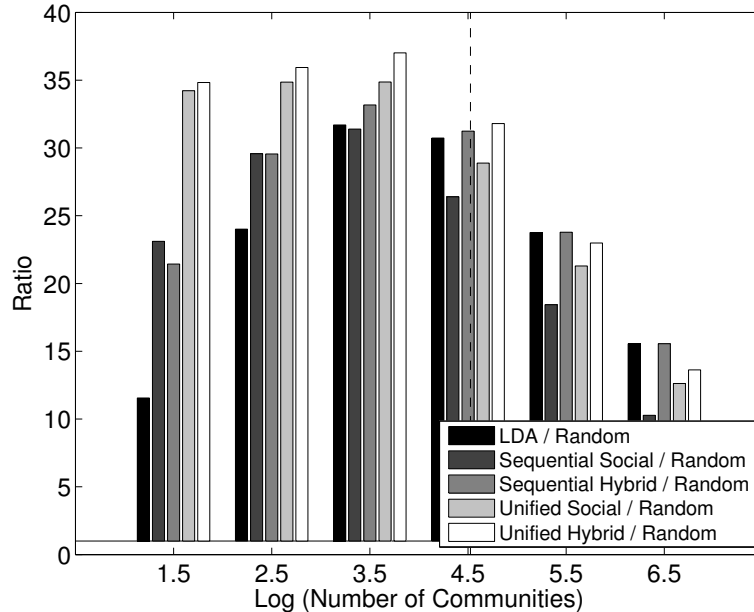


Figure 11: LiveJournal: AUC Ratio vs Log (# Communities)

*Model* that learns both in a unified way. To solve these models, we propose efficient parameter estimation solutions based on Expectation-Maximization that scale with the number of observed links. Our experiments with Epinions and LiveJournal data sets show that *Unified* outperforms *Sequential*, and both outperform the approach based on latent factors only (LDA).

## Acknowledgments

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

## References

- [1] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, June 2008.
- [2] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD ’06*, pages 44–54, New York, NY, USA, 2006. ACM.
- [3] R. Balasubramanyan and W. W. Cohen. Block-LDA: Jointly modeling entity-annotated text and entity-entity links. In *SDM*, 2011.
- [4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

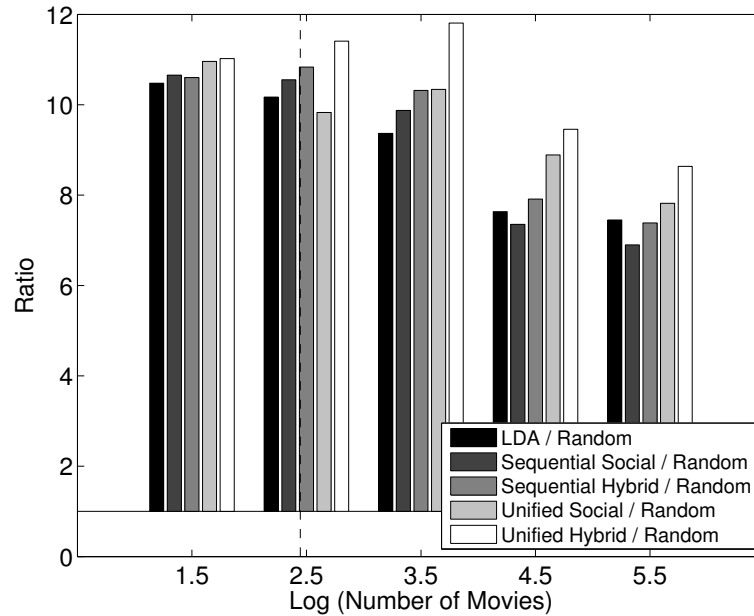


Figure 12: Epinions: AUC Ratio vs Log (# Movies)

- [5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [6] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '10*, pages 1029–1038, New York, NY, USA, 2010. ACM.
- [7] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '09*, pages 199–208, New York, NY, USA, 2009. ACM.
- [8] Dan Cosley, Daniel P. Huttenlocher, Jon M. Kleinberg, Xiangyang Lan, and Siddharth Suri. Sequential influence models in social networks. In *International Conference on Weblogs and Social Media*, 2010.
- [9] David Crandall, Dan Cosley, Daniel Huttenlocher, Jon Kleinberg, and Siddharth Suri. Feedback effects between similarity and social influence in online communities. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '08*, pages 160–168, New York, NY, USA, 2008. ACM.
- [10] Peng Cui, Fei Wang, Shaowei Liu, Mingdong Ou, Shiqiang Yang, and Lifeng Sun. Who should share what?: item-level social influence prediction for users and posts ranking. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, SIGIR '11*, pages 185–194, New York, NY, USA, 2011. ACM.

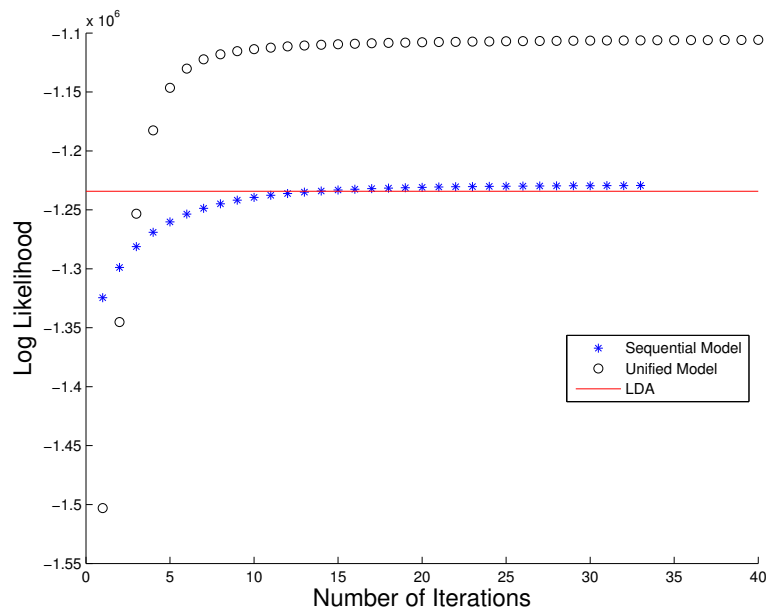


Figure 13: LiveJournal: Log Likelihood vs Number of Iterations

- [11] 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.
- [12] 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.
- [13] Amit Goyal, Francesco Bonchi, and Laks V.S. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining, WSDM '10*, pages 241–250, New York, NY, USA, 2010. ACM.
- [14] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.
- [15] Daniel Gruhl, R. Guha, David Liben-Nowell, and Andrew Tomkins. Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web, WWW '04*, pages 491–501, New York, NY, USA, 2004. ACM.
- [16] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '03*, pages 137–146, New York, NY, USA, 2003. ACM.
- [17] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '08*, pages 426–434, New York, NY, USA, 2008. ACM.

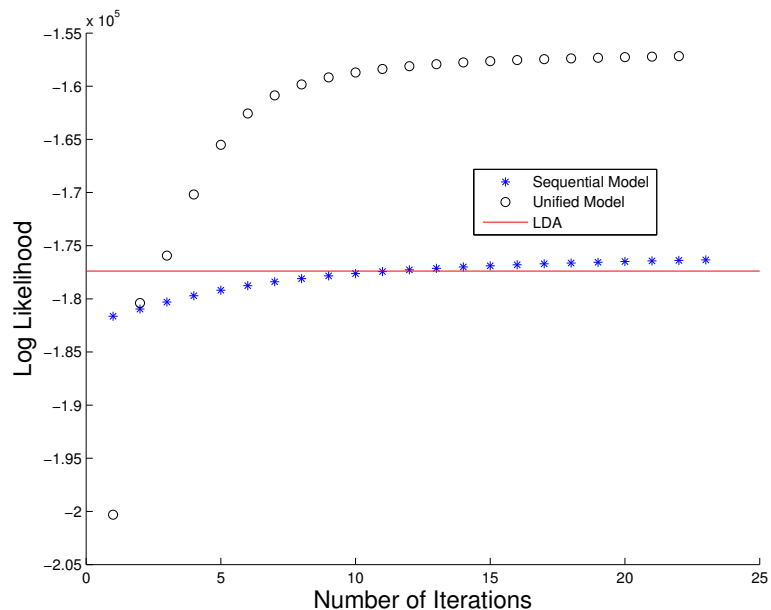


Figure 14: Epinions: Log Likelihood vs Number of Iterations

- [18] Yehuda Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data*, 4(1):1:1–1:24, January 2010.
- [19] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.
- [20] Timothy La Fond and Jennifer Neville. Randomization tests for distinguishing social influence and homophily effects. In *Proceedings of the 19th international conference on World wide web, WWW '10*, pages 601–610, New York, NY, USA, 2010. ACM.
- [21] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), May 2007.
- [22] Lu Liu, Jie Tang, Jiawei Han, Meng Jiang, and Shiqiang Yang. Mining topic-level influence in heterogeneous networks. In *Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM '10*, pages 199–208, New York, NY, USA, 2010. ACM.
- [23] Hao Ma, Irwin King, and Michael R. Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, SIGIR '09*, pages 203–210, New York, NY, USA, 2009. ACM.
- [24] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management, CIKM '08*, pages 931–940, New York, NY, USA, 2008. ACM.

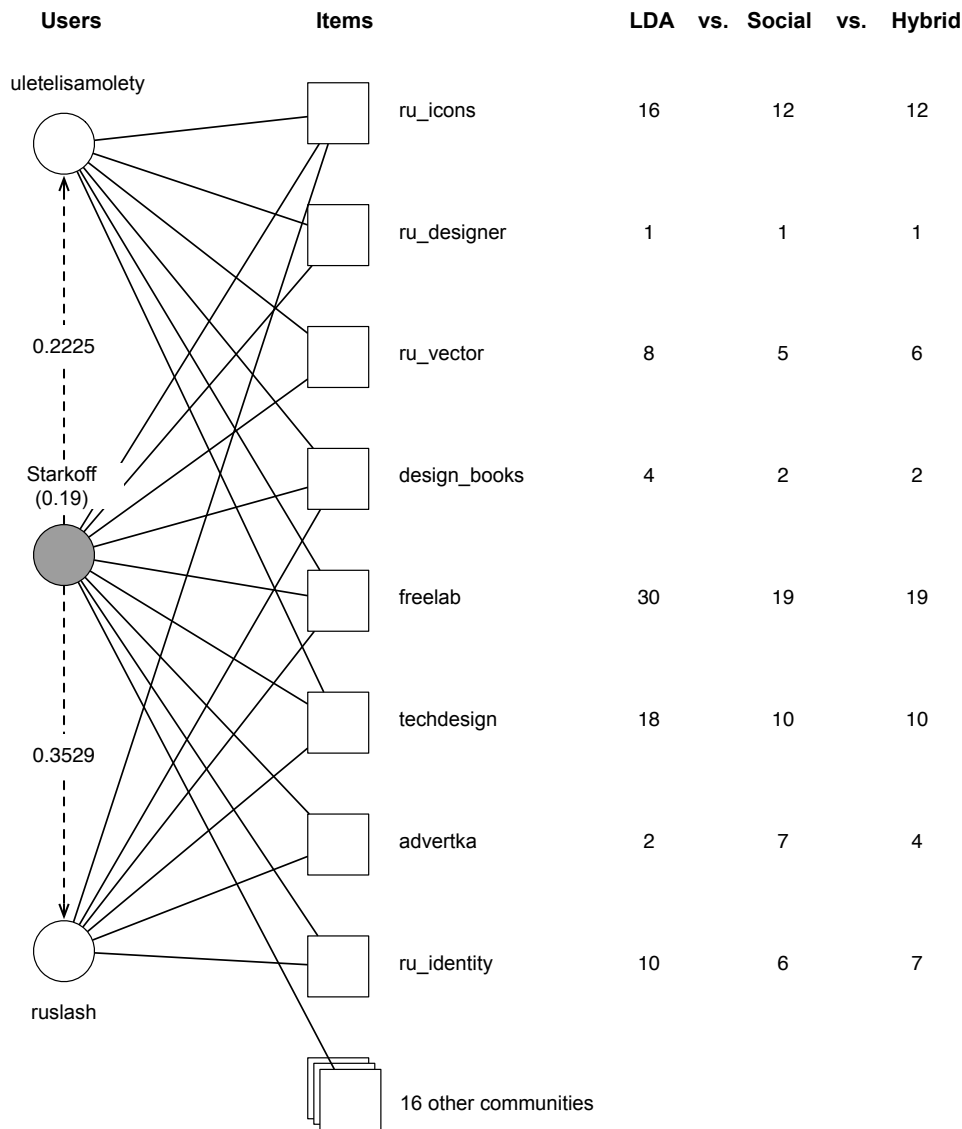


Figure 15: LiveJournal: Low Self Dependency

- [25] 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.
- [26] J. Neville and D. Jensen. Relational dependency networks. *J. Mach. Learn. Res.*, 8:653–692, 2007.
- [27] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, Cambridge, MA, 2008. MIT Press.
- [28] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning, ICML '08*, pages 880–887, New York, NY, USA, 2008. ACM.



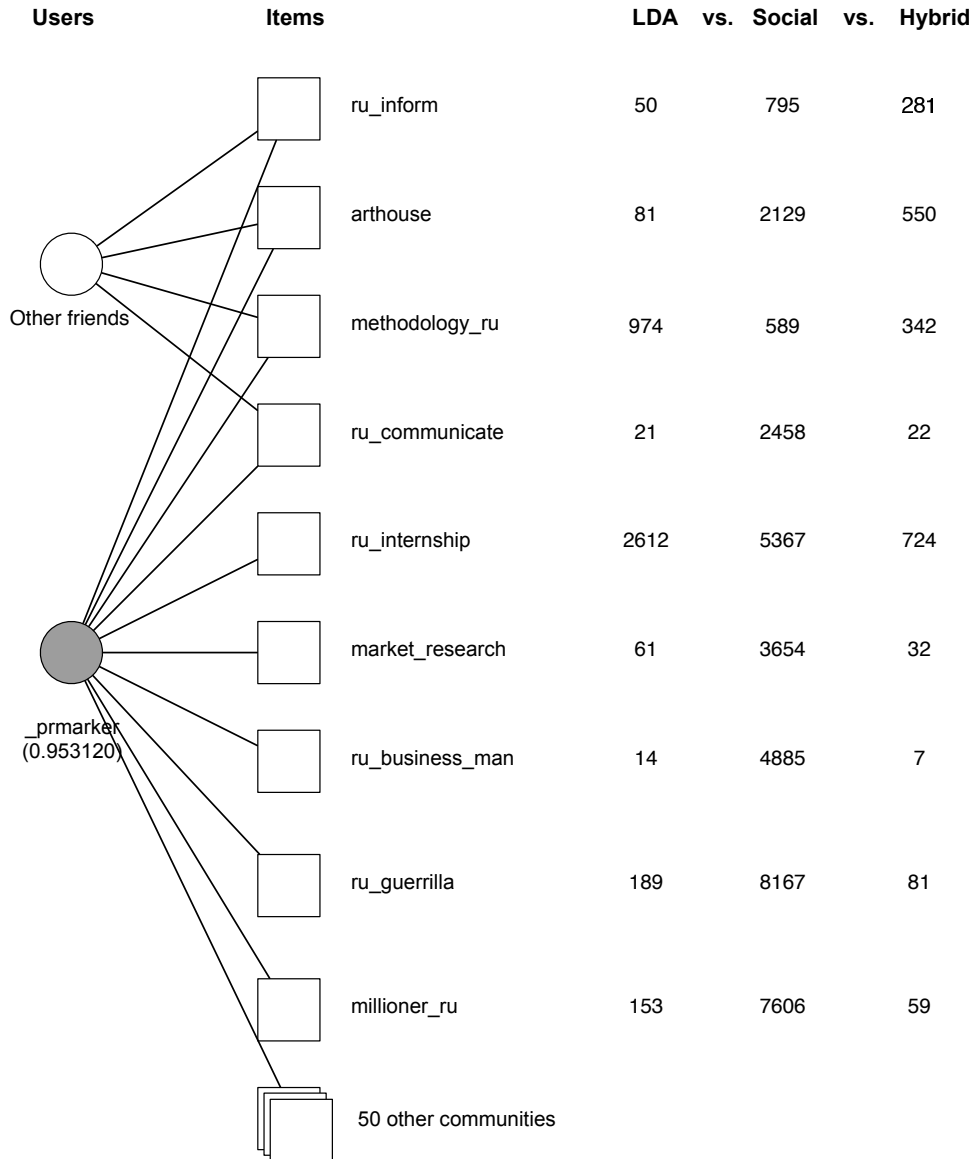


Figure 16: LiveJournal: High Self Dependency

- [29] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web, WWW '01*, pages 285–295, New York, NY, USA, 2001. ACM.
- [30] Parag Singla and Matthew Richardson. Yes, there is a correlation: - from social networks to personal behavior on the web. In *Proceedings of the 17th international conference on World Wide Web, WWW '08*, pages 655–664, New York, NY, USA, 2008. ACM.
- [31] Tristan Mark Snowsill, Nick Fyson, Tijl De Bie, and Nello Cristianini. Refining causality: who copied from whom? In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '11*, pages 466–474, New York, NY, USA, 2011. ACM.

- [32] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 448–456, New York, NY, USA, 2011. ACM.
- [33] Zhen Wen and Ching-Yung Lin. On the quality of inferring interests from social neighbors. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 373–382, New York, NY, USA, 2010. ACM.
- [34] Jaewon Yang and J. Leskovec. Modeling information diffusion in implicit networks. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 599–608, 2010.
- [35] Yu Zhang, Bin Cao, and Dit-Yan Yeung. Multi-domain collaborative filtering. In *UAI 2010*, pages 725–732, Corvallis, Oregon, 2010. AUAI Press.

## A Derivation of the E-Steps and M-Steps for Unified Generative Model

Suppose we have  $\Theta$  the users latent factor distributions and  $\Phi$  the latent factors item distribution. Then the likelihood of  $E$  is given by,

$$\begin{aligned} P(E|\Theta, \Phi, C, F) &= \prod_{u \in U} \prod_{v \in V_u} P(e_{v,u}|\Theta, \Phi, C, F) \\ &= \prod_{u \in U} \prod_{v \in V_u} \sum_{z \in Z} \sum_{x \in F_u} \left[ P(e_{v,u}|z_{v,u} = z, \Phi) \right. \\ &\quad \left. P(z_{v,u} = z|x_{v,u} = x, \Theta)P(x_{v,u} = x|C_u, F_u) \right] \end{aligned}$$

Then expressing in logarithm form,

$$\begin{aligned} \log P(E|\Theta, \Phi, C) &= \sum_{u \in U} \sum_{v \in V_u} \log \left[ \sum_{z \in Z} \sum_{x \in F_u} P(e_{v,u}|z_{v,u} = z, \Phi) \right. \\ &\quad \left. P(z_{v,u} = z|x_{v,u} = x, \Theta)P(x_{v,u} = x|C_u, F_u) \right] \end{aligned}$$

Find the E Step for  $z_{v,u}$  assuming that we do not have  $x_{v,u}$ ,

$$\begin{aligned} P(z_{v,u} = z|e_{v,u}, \Theta, \Phi, C, F) &= \frac{\sum_{x \in F_u} P(e_{v,u}, z, x_{v,u} = x|\Theta, \Phi, C, F)}{\sum_{z' \in Z} \sum_{x' \in F_u} P(e_{v,u}, z', x_{v,u} = x'|\Theta, \Phi, C, F)} \\ &\propto \sum_{x \in F_u} P(e_{v,u}|z, \Phi)P(z|x, \Theta)P(x|C_u, F_u) \\ &= g(u, z, v) \end{aligned}$$

Then find the E Step for  $x_{v,u}$  assuming that we do not have  $z_{v,u}$ ,

$$\begin{aligned} P(x_{v,u} = x | e_{v,u}, \Theta, \Phi, C, F) &= \frac{\sum_{z \in Z} P(e_{v,u}, z_{v,u} = z, x | \Theta, \Phi, C, F)}{\sum_{z' \in Z} \sum_{x' \in F_u} P(e_{v,u}, z_{v,u} = z', x' | \Theta, \Phi, C, F)} \\ &\propto \sum_{z \in Z} P(e_{v,u} | z, \Phi) P(z | x, \Theta) P(x | C_u, F_u) \\ &= h(u, x, v) \end{aligned}$$

In the M Step of EM algorithm, take partial derivative of the log likelihood with respect to  $\Theta, \Phi$  and  $C$ ,

$$\log P(E | \Theta, \Phi, C) = \sum_{u \in U} \sum_{v \in V_u} \log \left( \sum_{z \in Z} \sum_{u' \in U} \phi_{z,v} \theta_{u',z} c_{u,u'} \right)$$

Given that  $\sum_{u' \in U} c_{u,u'} = 1$ ,  $\sum_{z \in Z} \theta_{u,z} = 1$  and  $\sum_{v \in V_u} \phi_{z,v} = 1$  are constraints, we may optimize for the above using the following Lagrange constraint,

$$\begin{aligned} \mathcal{L}(\Theta, \Phi, C, F, \lambda) &= \log P(E | \Theta, \Phi, C, F) \\ &- \sum_{u \in U} \left[ \lambda_u \left( \sum_{x \in F_u} c_{u,x} - 1 \right) + \gamma_u \left( \sum_{z \in Z} \theta_{u,z} - 1 \right) \right] - \sum_{z \in Z} \delta_z \left( \sum_{v \in V_u} \phi_{z,v} - 1 \right) \end{aligned}$$

Suppose we differentiate  $\mathcal{L}(\Theta, \Phi, C, F, \lambda)$  with respect to  $c_{u,x}$ ,  $\theta_{u,z}$  and  $\phi_{z,v}$ :

$$\begin{aligned} \frac{d}{d c_{u,x}} \mathcal{L}(\Theta, \Phi, C, \lambda) &= \sum_{v \in V_u} \frac{\sum_{z \in Z} \phi_{z,v} \theta_{x,z}}{\sum_{z' \in Z} \sum_{x' \in F_u} \phi_{z',v} \theta_{x',z'} c_{u,x'}} - \lambda_u \\ \frac{d}{d \theta_{u,z}} \mathcal{L}(\Theta, \Phi, C, \lambda) &= \sum_{v \in V_u} \frac{\phi_{z,v} c_{u,x}}{\sum_{z' \in Z} \sum_{x' \in F_u} \phi_{z',v} \theta_{x',z'} c_{u,x'}} - \gamma_u \\ \frac{d}{d \phi_{z,v}} \mathcal{L}(\Theta, \Phi, C, \lambda) &= \sum_{u \in U} \frac{\sum_{x \in F_u} \theta_{x,z} c_{u,x}}{\sum_{z' \in Z} \sum_{x' \in F_u} \phi_{z',v} \theta_{x',z'} c_{u,x'}} - \delta_z \end{aligned}$$

Then find the  $c_{u,x}$ ,  $\theta_{u,z}$  and  $\phi_{z,v}$  which gives zero gradient for  $\mathcal{L}(C, \lambda)$ . To summarize, the E Steps are

$$\begin{aligned} f(u, v, z) &= \frac{\phi_{z,v} \theta_{u,z} c_{u,u}}{\sum_{z' \in Z} \sum_{x' \in F_u} \phi_{z',v} \theta_{x',z'} c_{u,x'}} \\ g(u, v, z) &= \frac{\sum_{x \in F_u} \phi_{z,v} \theta_{x,z} c_{u,x}}{\sum_{z' \in Z} \sum_{x' \in F_u} \phi_{z',v} \theta_{x',z'} c_{u,x'}} \\ h(u, v, x) &= \frac{\sum_{z \in Z} \phi_{z,v} \theta_{x,z} c_{u,x}}{\sum_{z' \in Z} \sum_{x' \in F_u} \phi_{z',v} \theta_{x',z'} c_{u,x'}} \end{aligned}$$

The M Steps are,

$$\begin{aligned} \theta_{u,z} &= \frac{1}{\gamma_u} \sum_{v \in V_u} f(u, v, z) \\ \phi_{z,v} &= \frac{1}{\delta_z} \sum_{u \in U} g(u, v, z) \\ c_{u,x} &= \frac{1}{\lambda_u} \sum_{v \in V_u} h(u, v, x) \end{aligned}$$

## B 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 LiveJournal and Epinions. For each topic, we identify the top items with the highest latent factor values for that topic.

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 6 shows a sample of the top movie

Table 5: Example Top Communities for Each Topic in LiveJournal

| <i>Topic L1</i> | <i>Topic L2</i> | <i>Topic L3</i> |
|-----------------|-----------------|-----------------|
| 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       |

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 6: Example Top Movie Titles for Each Topic in Epinions

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

## C Distribution of Social Correlation

Figures 17 and 18 show the histogram of self-dependency values. The x-axis indicates the self-dependency values in logarithm scale and y-axis indicates

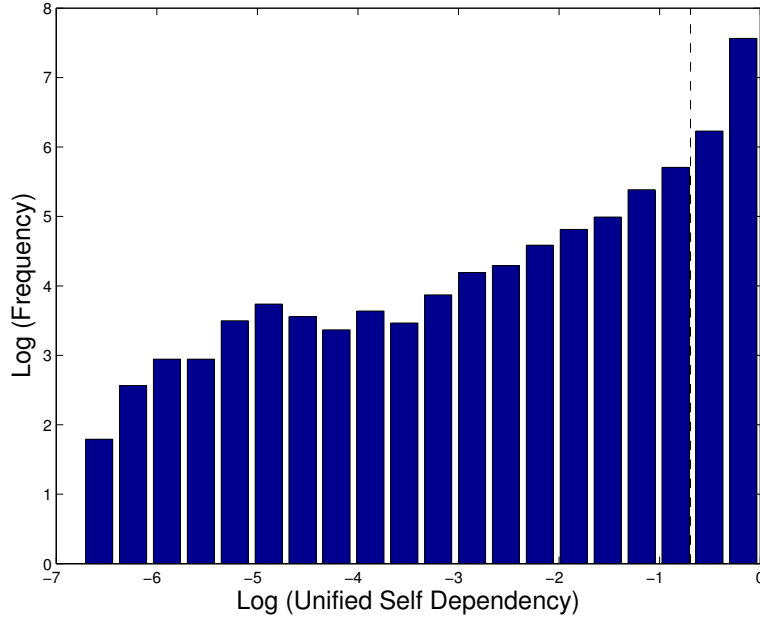


Figure 17: LiveJournal: Histogram of Self Dependency

the number of users who fall into the respective bins. The dotted black line parallel to the y-axis represents the logarithm value of 0.5. We define users having self dependency value less than 0.5 as followers (left of the dotted line), because they depend more on others in aggregate than in themselves. With this definition, 35% of users in LiveJournal and 29% of users in Epinions are followers. These significant percentages indicate that a sizable portion of the population do depend on others in their item adoptions, which validate our proposed approach of not relying on self preferences alone.

On the other hand, since the majority of users are non-followers, many social links between the users have very low social correlation values. In other words, a user may choose to follow another user but many of such follow relationships do not share common interests or result in item adoptions for the following user. This may imply that while the observed social network is sparse, the actual underlying dependency network between users is sparser.

## D Theoretical Performance of Random

Given that there are  $M$  items for the Random prediction model to select from and  $v$  out of  $M$  items are Actual Positive. That is, a random user has these  $v$  items in the testing set and we want to test how well Random method recovers these  $v$  items. Then given that we select the top  $k$  items returned by the Random method such that  $k \leq M$ . What is the probability that there are  $t$  correctly chosen items, given that  $t \leq v$ ?

Since AUC of Precision & Recall (AUC-PR) Curve for Random depends on the precision ( $PREC$ ) and recall ( $REC$ ) for each  $k$ , we should find the expected precision  $E(PREC|k)$  and expected recall  $E(REC|k)$  for each  $k$ . Expected values of precision and recall depends on the number of true positives ( $tp$ ) at  $k$ ,

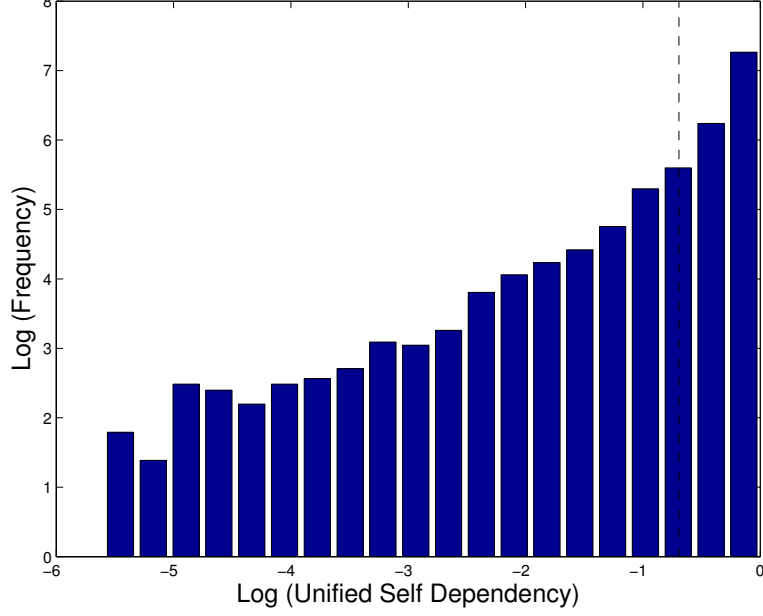


Figure 18: Epinions: Histogram of Self Dependency

$$\begin{aligned}
 E(PREC|k) &= \frac{E(tp|k)}{k} \\
 E(REC|k) &= \frac{E(tp|k)}{v} \\
 E(tp|k) &= \sum_{t=1}^{\min(k,v)} t \cdot P(tp = t|k) \\
 P(tp = t|k) &= \binom{v}{t} \cdot \binom{M-v}{k-t} / \binom{M}{k}
 \end{aligned}$$

$P(tp = t|k)$  is derived as follows, given that there are  $v$  actual positives, the number of possible ways to get  $t$  predicted positives, is the combinatorial  $\binom{v}{t}$ . Then there are  $M - v$  actual negatives, to select  $k - t$  predicted negatives out of these actual negatives, we have  $\binom{M-v}{k-t}$  different combinations of selections. Finally, there are  $\binom{M}{k}$  ways of choosing top  $k$  randomly from the entire possible set of items.

$P(tp = t|k)$  is in fact a HyperGeometric Distribution. Finally, expected AUC of PR Curve is given by the area under curve of the list of PR values for each  $k$ , from 1 to  $M$ .

Figure 19 shows the theoretical and actual empirical results given by *Random*. The performance of *Random* increases as number of items increases. This explains why our AUC ratio which represents the improvement over *Random* decreases when number of items increases, as shown in Figures 11 and 12. The values of AUC on the y-axis in Figure 19 shows that the AUC values are in the order of  $e^{-10}$  to  $e^{-3}$ . In comparison, the AUC values obtained by our models as reflected in Figures 5 and 6 are relatively higher than *Random*.

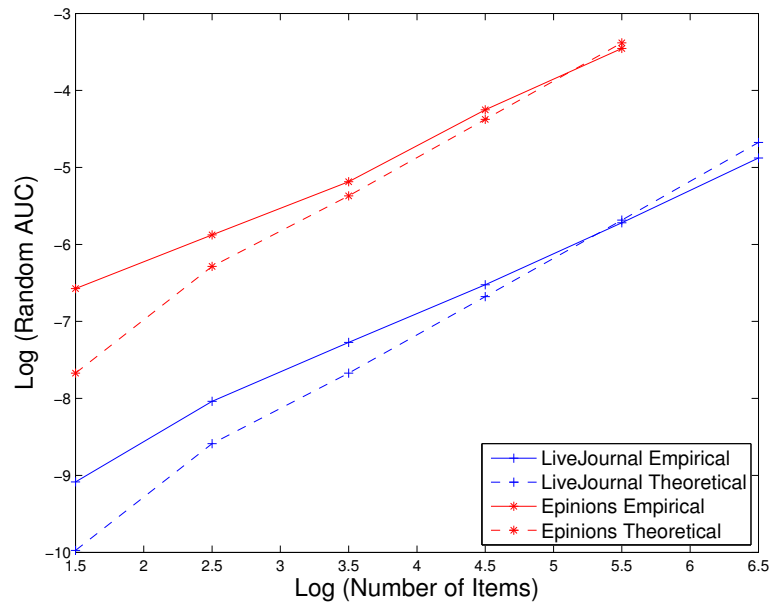


Figure 19: Log(AUC of Random) vs Log(Number of Items)