2-2014

Social Correlation in Latent Spaces for Complex Networks

Freddy Chong Tat CHUA
Singapore Management University, freddy.chua.2009@smu.edu.sg

Social Correlation in Latent Spaces for Complex Networks

FREDDY CHONG TAT CHUA

SINGAPORE MANAGEMENT UNIVERSITY

2013

This page was intentionally left blank.

Social Correlation in Latent Spaces for Complex Networks

by

Freddy Chong Tat Chua

Submitted to School of Information Systems in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Information Systems

Dissertation Committee:

Ee-Peng Lim (Supervisor/Chair)
Professor of Information Systems
Singapore Management University

David Lo
Assistant Professor of Information Systems
Singapore Management University

Archan Misra
Associate Professor of Information Systems
Singapore Management University

William Cohen
Professor of Machine Learning
Carnegie Mellon University

Singapore Management University
2013

Social Correlation in Latent Spaces for Complex Networks

by

Freddy Chong Tat Chua

Abstract

This dissertation addresses the subject of measuring social correlation among users within a complex social network. Social correlation is closely related to the measurement of social influence in social sciences. While social influence focuses on the existence of causal influence among users, we take a computational approach to measure correlation strength among users based on their shared interactions. We call this social correlation.

To formally model social correlation, we propose a framework which contains two major parts. The first part is that of representing users behavior in a computationally efficient and accurate manner. For example, social media users perform many kinds of actions online such as buying products, watching videos and posting comments. The huge number of users' actions logged over long duration poses significant challenges for analysis. We propose both static and temporal models to compress the huge amounts of users' action data into low dimensional representations. For the dynamic users' action data, there is the additional challenge of temporal sparsity where users have low amounts of activities in some time periods. This results in the lack of information in some time periods for modeling the temporal behavior of users. By exploiting the transition of users behavior in different time periods, we obtained a smoothed representation of users behavior in low dimensions.

The second part of modeling social correlation is to take the users' behavior in low dimensional representation and compare against the behaviors of other users whom they had earlier interacted with. The dissertation first proposes social correlation measurement for the static representation of users' behavior.

It then extends the measurement to the temporal case by using only two time periods and finally for the general case of multiple time periods using Granger causality. With our proposed set of social correlation measurements one can now build better recommendation systems that predict the missing or future users' behavior considering the influence among users in the complex networks.

Contents

List of Figures

This page was intentionally left blank.

List of Tables

This page was intentionally left blank.

Publications based on the Dissertation

Listed in reverse chronological order:

1. Freddy Chong Tat Chua, Richard J. Oentaryo and Ee-Peng Lim. Temporal Social Correlation Using Linear Dynamical Topic Model with Granger Causality. *Working paper*. (Chapter 6)

2. Freddy Chong Tat Chua, Richard J. Oentaryo and Ee-Peng Lim. Modeling Temporal Adoptions Using Dynamic Matrix Factorization. *Proceedings of the IEEE International Conference on Data Mining*, (ICDM 2013). (Chapter 5)

3. Freddy Chong Tat Chua, Hady W. Lauw and Ee-Peng Lim. Generative Models for Item Adoptions Using Social Correlation. *IEEE Transactions on Knowledge and Data Engineering*, (TKDE 2013). (Chapter 3)

4. Freddy Chong Tat Chua, Hady W. Lauw and Ee-Peng Lim. Mining Social Dependencies in Dynamic Interaction Networks. *Proceedings of the SIAM International Conference on Data Mining*, (SDM 2012). (Chapter 4)

5. Freddy Chong Tat Chua, Hady W. Lauw and Ee-Peng Lim. Predicting Item Adoption Using Social Correlation. *Proceedings of the SIAM International Conference on Data Mining*, (SDM 2011). (Chapter 3)

This page was intentionally left blank.

Acknowledgements

The completion of this dissertation is made possible by the assistance given to me by several persons whom I am very much grateful to. It would be hard to quantify the amount of benefit I have received from them, and much more complicated to measure how those benefits compounded over the years to help me achieve what I have today. The least I could do is to acknowledge them in this short little section and to hope I may one day be like them by returning those benefits back to society or anyone else who needs it in the best of my abilities.

First of all, I feel fortunate to live in an era of abundance and a relatively wealthy country, where scientific and technological progresses have provided productivity gains to free up human capital for academic pursuits. The abundance of our society have created sufficient wealth to support a small group of people which includes me, to participate in the activities of academic research, that does not require us to immediately generate any tangible economic returns to our society. I could then take a long-term view and develop fundamental skills to address the needs of our society in the longer term.

I was introduced to Professor Lim Ee-Peng by Dr. Liu Zehua in late 2008 after the onset of the Global Financial Crisis, considered by many economists to be the worst financial crisis since the Great Depression of the 1930s. I formally started my Ph.D. programme in 2009 under the supervision of Professor Lim Ee-Peng, who has provided a stable, conducive and intellectual environment for me to develop my research abilities. I am grateful to Professor Lim Ee-

Peng, Professor Pang Hwee Hwa and Professor Steven Miller for accepting me into their Ph.D. programme. I am thankful to Professor Lim Ee-Peng, Professor Steven Miller and Professor Stephen E. Fienberg for establishing the Living Analytics Research Centre (LARC), which provided very life-altering opportunities for my research career. Under the scholarship from LARC, I have the opportunity of working with Professor William W. Cohen from Carnegie Mellon University and subsequently undertake a summer internship with Dr. Bernardo A. Huberman's Social Computing Group.

During my candidature, I have had the fortune of working with several brilliant researchers. In the early stages, I worked with Dr. Hady W. Lauw on the foundations of my dissertation topic, which led to the publication of three research papers. After my return to Singapore, Dr. Richard J. Oentaryo gave me helpful advice in the research that completes the rest of my dissertation and led to two research papers. Professor William W. Cohen supervised me when I visited Carnegie Mellon University (CMU) in 2011 and we co-authored a paper together. Dr. Sitaram Asur was the scientist who supervised me during the internship at Hewlett-Packard Research Labs during the summer of 2012 and we co-authored a paper as well as applied for a patent.

During my stay at Pittsburgh, Pennsylvania, USA from August 2011 until May 2012, I am glad to have the company of the few persons, Zhang Jilian, Li Yan, Tang Yanming and Yan Qiang.

I will like to thank the support of several scientists, fellows, engineers and students: Dai Bing Tian, Gao Ming, Palakorn Achananuparp, Philips K. Prasetyo, Agus T. Kwee, Ibrahim Nelman Lubis, David Low, Luu Minh Duc, Hoang Tuan Anh and Gong Wei.

I appreciate the comments and feedback my dissertation committee members have given me: Professor Ee-Peng Lim, Professor William W. Cohen, Associate Professor Archan Misra and Assistant Professor David Lo.

The following persons have written recommendation letters for me: Profes-

sor Lim Ee-Peng, Professor William W. Cohen, Consulting Professor Bernardo A. Huberman, Professor Jaideep Srivastava, Professor Michael W. Macy, Professor Stephen E. Fienberg, Associate Professor Lau Hoong Chuin, Assistant Professor Hady W. Lauw, Assistant Professor Feida Zhu, Dr. Jiang Daxin, Dr. David Woon, Associate Professor Tan Tiow Seng, Associate Professor Abhik Roychoudhury and Dr. Gary Lee.

I also want to commend on the excellent administrative support given to me by the staffs of LARC; Chua Kian Peng, Alenzia Wong Poh Luan, Angela Kwek Renfeng, Fong Soon Keat, Phoebe Yeo, Desmond Yap, Nancy Beatty, Ashley Ferenczy and the staffs from School of Information Systems; Ong Chew Hong, Seow Pei Huan and many others whom I have no opportunity to know them better.

The research leading to the completion of this dissertation is funded by,

1. The Singapore National Research Foundations Research Grant, NRF2008IDM-IDM004-036.

2. The Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

This page was intentionally left blank.

Dedicated to Mummy and Papa.

This page was intentionally left blank.

Chapter 1

Introduction

Unprecedented progress and innovation on web and mobile technologies provide consumers with a wide variety of choices. Today, online shopping provides access to many different consumer items such as books, cameras and movies 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. Users need an intelligent system that understands their personal preferences based on the social communities they belong to.

In making purchase or adoption decisions, users rely on the social communities to organize the complex information and content related to consumer items 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 become a conduit for discovering relevant information. In platforms such as Twitter and Epinions, users can choose 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.

Social Correlation can be useful in many different applications including diffusion of innovations, viral marketing, recommendation of new products, measurement of influential users, and prediction of item adoptions, etc. [10, 11, 64, 74, 80, 92, 114]. The strength of social correlation links also allows us to determine the cohesiveness of users, which can be used to divide users into smaller communities [65, 77, 106]. Another related concept is that of social influence which is commonly investigated by social scientists who are concerned about proving the existence of causality between the actions of users. However, in social correlation, we are concerned about the issues related to computational efficiency and application of social correlation for predictions or recommendations.

We use the term "users adopting items" to describe any action of a user that captures her preferences. Examples of users adopting items include users watching movies, users joining online communities [27, 29] and users producing text content [28, 30]. This is also the most common type of user actions in the users' item adoption data that is commonly available, where items may refer to any form of media.

In this dissertation, we propose a framework that allows us to compute the social correlation between users based on their adoption behavior. The framework as shown in Figure 1.1, requires two major parts which is addressed in the following chapters.

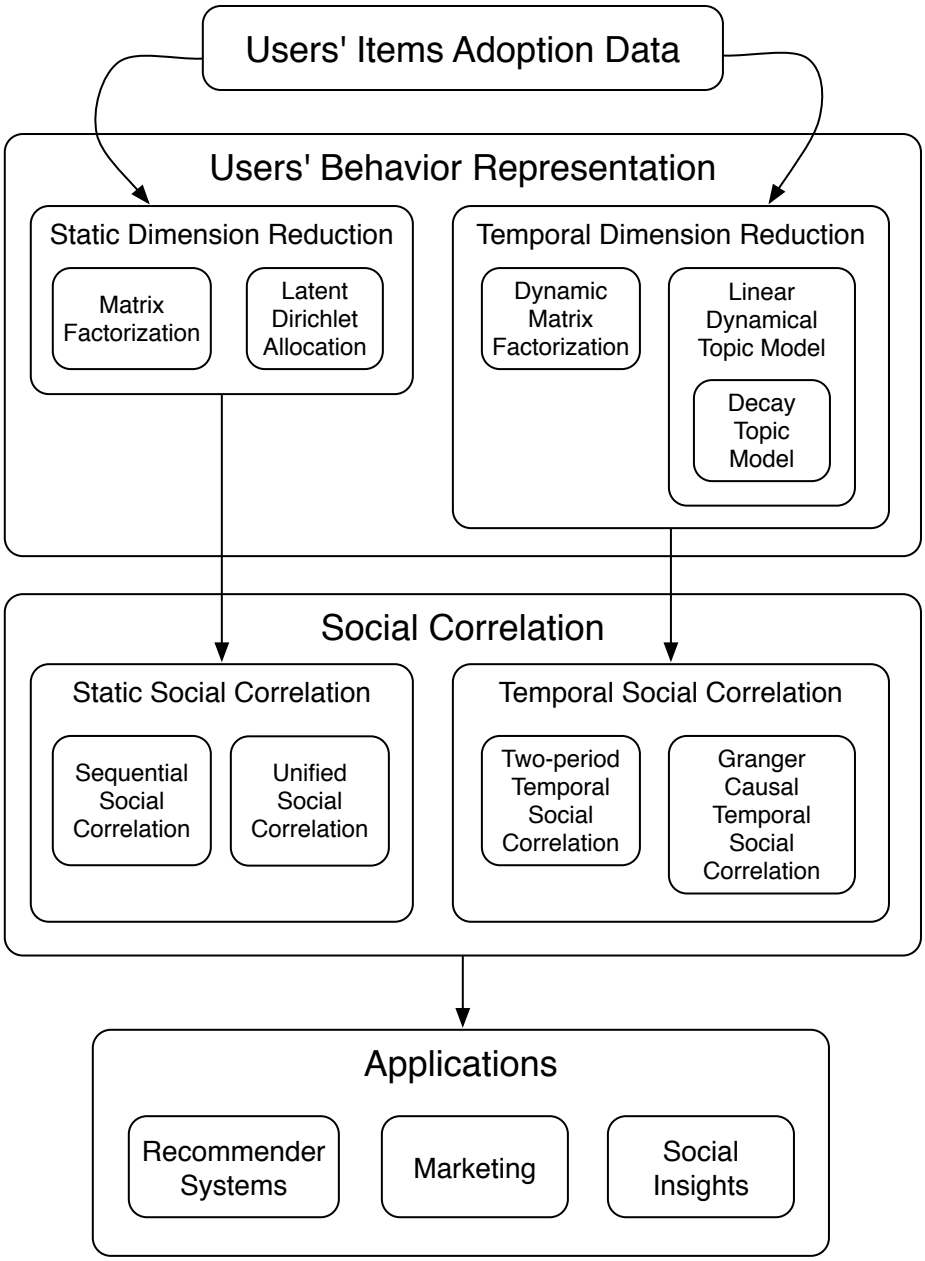


Figure 1.1: Social Correlation Framework

The first step requires us to obtain the adoption behavior of users as a vector with manageable dimensions. For example, a user could choose to adopt M different kinds of items but we will need to represent a user's adoption using only K dimensions where K is significantly smaller than M. The second step for computing social correlation is to use the previously computed adoption vectors as an input to various social correlation measures.

In each of these two steps, we propose new models that allow us to analyze

both static and temporal data sets. Due to the generality of this framework, one could combine different combinations of dimension reduction methods with the social correlation measures. The social correlation values could then be used for various applications in recommender systems, marketing of products or derive social insights from the social data.

1.1 Users' Behavior Representation

In users' item adoption data, the number of possible items for users to adopt is very large and can be in the order of millions. A naive way of representing users' behavior is to use a vector with dimensions similar to the number of possible items, with entries representing the raw frequencies of adoption. Unfortunately, using high dimensional vectors for users' behavior would lead to expensive computations during the comparison between users. Comparing only the raw frequencies would also be less informative since we ignore the co-occurrence relationships between different items adopted by the users. A widely adopted method of reducing the computational costs is to perform dimension reduction on the users' item adoption data to obtain vectorized representations in lower dimensions such as order of tens or hundreds. Various methods exist for performing dimension reductions on static and temporal data sets.

Static Dimension Reduction:

1. Matrix Factorization (MF): Suppose the matrix $Y \in \mathbf{R}^{M \times N}$ represents the original users' items adoption data, where M represents the number of items and N represents the number of users [89, 90]. By applying dimension reduction on Y, we obtain $C \in \mathbf{R}^{M \times K}$ and $X \in \mathbf{R}^{K \times N}$, where K is significantly smaller than M or N. Each row in C represents the item's latent factor while each column in X represents the user's latent factor. We use the users' latent factors as the compressed vectorized

representations of the users' behavior.

2. Non-negative Matrix Factorization (NMF): The NMF [62, 63] is a direct extension of MF by enforcing non-negativity constraints on the entries of the derived items' and users' factor matrices. There are qualitative benefits for using NMF especially when we need to understand the semantic meaning of items' latent factors.

3. Latent Dirichlet Allocation (LDA): LDA [16] was derived from the intersection of Natural Language Processing and Bayesian Machine Learning. In LDA, text documents are treated as a bag of words. Each document is associated with a topic distribution and each word is associated with a latent topic variable. The value of the latent topic variable is inferred as a mixture of how likely the document will generate the topic and how likely the topic will generate the word. We could also use LDA as a dimension reduction method for users' item adoption data. Each user could be associated with a topic distribution indicating their interests of behavior in adopting items. Similarly, each item could be associated with a latent topic variable.

Temporal Dimension Reduction: Instead of only a single users items adoption matrix Y, we now have multiple matrices Y_t each at different time step t. A direct extension of static dimension reduction methods on temporal data is to apply any of those methods (MF, NMF, LDA) on each time step independently of other time steps. But due to the randomization effects of the MF, NMF and LDA algorithms and the temporal sparsity problem where some users have low or no activity in some periods, applying dimension reduction on each time step independently would lead to unrelated latent factors across the different time steps. In this dissertation, we develop three different models for representing users' behavior.

1. Decay Topic Model (DTM): Decay Topic Model [28] extends LDA to

obtain topic distributions of users at different time steps. The decay factor is used to balance between the information that we have learned about users in older time steps and the use of recent item adoptions to obtain an updated knowledge of users' behavior and preferences. In order to simplify the temporal probabilistic model, we have chose to fix the decay parameters.

2. Dynamic Matrix Factorization (DMF): DMF [30, 99] is the result of extending MF by using Linear Dynamical Systems (LDS) to linearly transform the users' behavior between time steps. The DMF approach provides the elegance of making less assumptions about how users evolve and allow the algorithm to automatically smoothen the dynamics of users' behavior between different time steps. However, a significant drawback of the DMF model is that we are not able to learn the items' latent factor, users' latent factor and dynamics matrix in an iterative unified manner.

3. Linear Dynamical Topic Model (LDTM): We propose LDTM to overcome the fixed decay constraints in DTM. We merge the use of DMF and LDA to obtain topic distributions of users at different time steps while allowing for the automatic estimation of the decay parameters for each user. The LDTM combines both benefits of DTM and DMF by estimating for all the parameters in an iterative and unified manner.

1.2 Social Correlation

On predicting whether a user will adopt an item, a dot product of user latent factor and item latent factor is sufficient to yield an estimate for the missing or sparse relationship in the original data. However, our aim in this dissertation is to draw upon the item adoptions to measure the amount of social correlation between users. The derived social correlation could help us determine the correlation of item adoptions between users, which could be used as an

additional feature for recommender systems. Similar to the users' behavior representation, we also develop different social correlation measures for both static and temporal data sets:

Static Social Correlation: The static social correlation measure is a quantity between every pair of users that optimizes the likelihood of observing the users item adoption data beyond non-social factors can capture. To model an observed count of adoptions for an item m by a user n, the non-social approach is to use only item m's latent factor and user n's latent factor, while the static social correlation approach is to use item m's latent factor, user n's latent factor as well as the latent factors of user n's friends. The extent user n would rely on each of her friends would depend on user n's static social correlation with each of them.

1. Sequential Static Social Correlation: The sequential model of static social correlation would first derive the items' latent factors and the users' latent factors using dimension reduction. After which we obtain the social correlation measures as an independent step from the previously obtained user latent factors. The static social correlation linearly combines the user's latent factors and her neighbors' latent factors to obtain a better quantitative explanation for the observation of item adoptions.

2. Unified Static Social Correlation: The unified model works on the same basic principle as the sequential model with the exception that we iteratively obtain better estimates of both latent factors and social correlation measures in a positive feedback loop. This cyclical approach allows us to obtain parameters that give better likelihood estimates of the observed data.

Temporal Social Correlation: A natural extension of static social correlation is to use the temporal users' behavior representation at every time step to derive the social correlation between users with dynamic behavioral data.

We propose to first use two consecutive time steps to obtain the temporal social correlation for a pair of users and further generalize the temporal social correlation model for a window of multiple periods.

1. Two-period Temporal Social Correlation: This approach uses the interactions of users to co-adopt items. The basic principle is similar to static social correlation where we use the users' and neighbors' latent factor in previous time step and optimize the social correlation to model the items adoptions in the next time step. Because of the adherence to previously defined static social correlation, the temporal social correlation here could only utilize the adoption information in only two consecutive time steps.

2. Granger-causal Temporal Social Correlation: Due to the limitations of our earlier definition for the Two-period Temporal Social Correlation, we propose to use Granger causality in order to generalize the temporal social correlation for a variable number of time steps. The linear regression models used in Granger causality also allows us to predict the future behavior of users based on the behavior of the users' neighbors.

1.3 Contributions

We make the following contributions in this dissertation:

1. We propose a Social Correlation Framework that incorporates social correlation in the generation of users' item adoptions for both static and temporal data sets. For static data sets, we propose novel static social correlation measures from the use of existing methods of static dimension reduction. For temporal data sets, we propose novel temporal dimension reduction techniques to use with existing causality measures.

2. In static social correlation, we propose two generative models: *Sequential*

Generative Model and *Unified Generative Model*. The Sequential Generative Model learns in two sequential steps, first employing LDA to learn the parameters of the user and item latent factors, followed by learning social correlation based on those parameters. The Unified Generative Model learns social correlation simultaneously with the user and item latent factors in a principled, 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).

3. In temporal social correlation, we model how a user's latent factors may change over time. Our proposed model, called *Decay Topic Model* (DTM), measures the personal topic preferences of a user at every time step. This model is novel in that unlike previous topic models (see Section 2.3) where documents have fixed topic distributions and only the topics may change, in our model users may have different affiliations to topics over time. Furthermore, a decay factor is included in the topic model to moderate the rate of change in topic preferences of users so as to create smooth transition of topic preferences as well as to address missing user interaction data.

4. Given the interaction links among users and topic preferences determined by decay topic model, we propose *Two-period Temporal Social Correlation* that measures how a user correlates with other users in producing or adopting content. The temporal social correlation is topic-based and it considers the topic preferences of a user in the current time step and other users' in the previous time step. This notion of changing temporal social correlation of a user that also takes into account the changing topic preferences of others that the user depends on, is a novel concept.

5. We propose Dynamic Matrix Factorization (DMF) based on Non-Negative Matrix Factorization (NMF) and Linear Dynamical Systems (LDS), and

apply it to solving several prediction tasks involving adoptions at different time steps as well cumulated adoptions across multiple time steps. We also derive a few variants of DMF based on the choice of item factor scaling and dynamics matrix and show how they can be used in the different adoption prediction tasks.

6. We propose a novel approach called Linear Dynamical Topic Model (LDTM) that merges the benefits of DMF and DTM. The proposed LDTM model represents user adoption behavior as topic distributions at different time steps and we track the evolution of users' topic distribution using Linear Dynamical System (LDS). By formulating the dynamics of parameters among different time steps using LDS, we obtain a dynamics matrix $A_{n,t}$ for each user n, which allows us to automatically decay their prior adoption behavior on every time step. To the best of our knowledge, such estimation of dynamics matrix $A_{n,t}$ has not been proposed in any topic models.

1.4 Organization of the Dissertation

The research written in this dissertation is an aggregation of several research papers we have written. We first discuss some research that are related to our topic in Chapter 2. Then we begin with our research on static data sets in Chapter 3 [27, 29] which require the use of static dimension reduction and static social correlation. Success on static data sets gave us confidence to progress on to dynamic data sets in Chapter 4 [28], where we explored the use of Decay Topic Model (DTM) and Two-step Temporal Social Correlation. Because of the deficiencies of DTM, we investigated the feasibility of Dynamic Matrix Factorization (DMF) on the users item adoption problem in Chapter 5 [30]. Chapter 5 also highlighted the differences between the users item adoption problem and users rating problem. Finally, we pool our accumulated knowledge

from the prior chapters to derive the Linear Dynamical Topic Model (LDTM) and Granger-causal Temporal Social Correlation in Chapter 6. Chapter 7 concludes this dissertation and set the stage for some promising future work.

This page was intentionally left blank.

Chapter 2

Related Work

We review some related concepts and prior work that motivate our research in this dissertation. We would first review the classical concept and existing research on social influence, followed by existing algorithms for performing dimension reduction on static and temporal data sets. We would then review existing works that focus solely on estimating the degree of causality in general time series data. We also include a section to mention some related works that has used some of these dimension reduction and influence concepts but in a manner that is significantly different from our approach. We end this chapter by highligting the main differences between our dissertation and the research in existing literature.

2.1 Social Influence

Social influence can take on different meanings depending on the context of discussion. We base our discussion of social influence on users' items adoption data while preserving the key concepts common to other domains. We say that user x socially influences user y only if user y adopts an item *because* user x has adopted the item. This simple statement has several implied conditions:

1. Temporal: It implies that we can observe user x has adopted the item **before** user y.

2. Interaction: User x has **interacted** with user y for y to know that x has adopted the item. y then follows the action of x by adopting the same item.

3. Confounding: User y has adopted the item **solely because** of x's adoption. The term *because* suggests a notion of causality while *solely* implies that no other factors *caused* the adoption. If other factors exist, then these factors will confound our belief in the presence of social influence between two users.

All of these conditions must be satisfied before we can conclude the presence of social influence between two users.

The *temporal* condition is simply an issue of data availability and given that we have arrived at the age of big data, it will be increasingly easy for us to meet this condition.

The condition of observing an *interaction* between two users is also an issue of data availability, that is, whether x has communicated with y through text messaging on social media platforms. Crandall et al. [32] used the discussion activities between Wikipedia articles to model the social interactions. However, due to privacy concerns, researchers often do not have access to the contents of commuication between users in social media. Most of the social influence research [5, 6, 7, 17, 61, 76] overcomes the lack of interaction data by assuming that users who are socially connected at the point of item adoption have interacted with one another. While not entirely true, this is generally accepted within academic research.

The most difficult condition to meet is that of eliminating all other external (confounding) variables that would confound our belief in the existence of influence between two users. That is, given that we have already satisfy the *temporal* and *interaction* conditions, we would have to prove that user y has not adopted the item because of the item's inherent attributes or attractiveness. For example, the presence of homophily between users would confound the

studies of social influence. Several studies [5, 6, 7, 32, 61, 76] have pointed out that socially connected users tend to have common item adoptions due to the effects of homophily; users with similar attributes or preferences tend to be socially connected and would adopt the same items regardless of any existence of social influence.

Fond and Neville [61] established that correlations between social connections and common item adoptions is a result of two processes that happen alternatively over a period of time: "homophily" causing users with similar attributes to form social connections, and "influence" causing users with social connections to become more similar in attributes.

To overcome the problem of confounding variables for proving the existence of social influence, researchers *randomly* group users into two groups. One group would receive the influence (treatment) while the other group functions as the control group that does not receive the influence (treatment). This *Randomization* technique is often used in physics to find the cause of a phenomenon and medicine to prove the effectiveness of a drug. The purpose of randomization is to minimize the effects of confounding variables present in every user. When randomized users are placed together in a group, the assumption is that the diversity of users' characteristics would collectively cancel-out the effects of their confounding variables on one another. The reliability of randomization goes beyond the scope of this dissertation but readers may refer to [46, 87] for more information.

The randomization technique is useful in cases where it is easy to manipulate which users would receive the treatment. Such users may belong to an online website where researchers have access to manipulate the website, or users may be recruited to participate in experiments on a voluntary basis. Aral and Walker created a Facebook application to test whether broadcast or personalized messages have any social influence effects [6] on the friends of recruited users. Bond et al. collaborated with Facebook Inc. and conducted experiments

on Facebook users to study whether online political messages could influence the voting decisions of users [17]. Muchnik et al. studied how the votes of news articles affected the articles' discussions [76].

When researchers do not have access to manipulate online web systems or recruit users, the randomization technique cannot be applied to obtain data for social influence analysis. An alternative to randomization is to perform *Quasi-Experiments* on a set of collected data to simulate effects of randomization [91]. A classic example of Quasi-Experiment is the *Matched Sampling* technique proposed by Rosenbaum and Rubin [85]. The Matched Sampling technique works by finding pairs of users with similar characteristics and applying the social influence (treatment) to one of the user in each pair, while the other user who does not receive the treatment functions as the control.

Aral et al. adapted the use Matched Sampling technique in Yahoo! Messenger data for distinguishing between influence and homophily in the adoption of a mobile service application (Yahoo! Go) [7]. Aral et al. first calculated the propensity of adoption based on the inherent characteristics of users, then matched pairs of users where one has adopter friends while the other has no adopter friends. In each pair, a user with adopter friends is said to have been socially influenced if her counterpart without adopter friends did not adopt the item even though both has similar propensity of adoption.

Anagnostopoulos et al. [5] proposed the *Shuffle Test* in order to distinguish influence from homophily. The Shuffle Test hypothesizes that the order of adoptions of a users' neighbors play a role in influencing whether a user eventually adopts the item. If shuffling the order of adoptions for a users' neighbor does not change the propensity of adoption, then it indicates that there is no influence that leads to the adoption of the item for the user.

To summarize, much of the Social Influence research has revolved around the following:

1. Adoption of a *single* item.

2. Satisfaction of the confounding condition; elimination of confounding variables for proving existence of causality.

The research we pursue in this dissertation is different in several ways. We utilize the set of items adopted by users as opposed to a single item. We propose algorithms and models to translate the high dimensional set of items users adopt into low dimensional representations for users' behavior and preferences. These algorithms and models work both on static and temporal data sets.

While existing works prove the existence of social influence in the adoption of item for users, the social influence is a discrete value to express the presence or absence. Our dissertation proposes methods to derive a correlation weight between every pair of users which determines how correlated their adoption behavior are over time.

2.2 Static Latent Factor Models

We discuss some existing latent factor models for static dimension reduction. There are two dominant families of latent factor models, Topic Models and Matrix Factorization (MF). Topic Models obtains the probabilistic latent factors by maximizing for the log-likelihood, while Matrix Factorization minimizes for the least squared error to obtain latent factors in the real-space. Because we use and extend Topic Models for our research in Chapters 3, 4 and 6, it would be necessary for our readers to have a basic understanding of Topic Models [9, 16, 47, 84] in the original formulation, Latent Dirichlet Allocation (LDA) [16]. Because Matrix Factorization and its extensions [62, 63, 89, 90] are used in our research in Chapters 5 and 6, we would also want our readers to have an overview of MF.

Topic Model

LDA was formerly conceived as a way of modeling unigram words in a document corpus [16]. 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 the distribution of topics. Figure 2.1 refers to the graphical notation of LDA. The generative process for LDA is as follows,

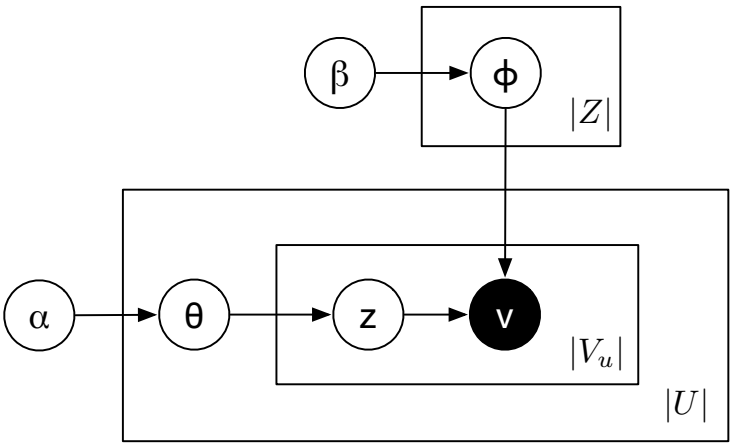


Figure 2.1: Latent Dirichlet Allocation in Plate Notation

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

$$\theta_u \sim Dirichlet(\alpha)$$

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

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

3. Then from the latent factors of item distributions Φ, u chooses the item

v from as follows:

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

where Φ follows a symmetric Dirichlet distribution with hyper-parameters β, as follows:

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

Solving for these parameters is fundamentally a likelihood optimization problem subjected to the probabilistic constraints. Blei et al. showed that the parameters can be estimated using variational expectation maximization [16], while Griffiths and Steyvers subsequently showed that LDA can be estimated easily using Gibbs Sampling [47].

But LDA only models the dyadic relationship between users and items. Several authors [9, 84] have extended LDA to relate the user - user relationships with users - items relationships. Balasubramanya and Cohen [9] had proposed Block-LDA that unifies the Mixed Membership Stochastic Blockmodels [4, 78] and LDA to jointly model the user to user relationships.

The Mixed Membership Stochastic Blockmodel (MMSB) proposed by Airoldi et al. [4] uses probability distributions to denote that a user could belong to a set of social communities each with varying degree of memberships. This is in contrast to the other community detection algorithms [66, 77] which assumes that each user only belongs to one community. But MMSB models the Bernuolli distribution of all N^2 observed and unobserved edges. As a result, MMSB is not able to utilize the sparse structure of social networks and could only be used to fit networks of smaller scale. To address the sparse structure of social networks, Parkkinen et al. [78] improvised by modeling the users at both ends of the edges using a Multinoimial distribution rather than the Bernuolli probability of whether the edge is observed. This allows for the social network to be modeled on the sparsedly observed data, which greatly improves the computation cost as compared to MMSB [4]. However, Blockmodels [4, 78] only

models user to user relationships and ignore the relationship between users and items.

Building on the sparse Blockmodel of Parkkinen et al. [78], Balasubramanya and Cohen [9] jointly model the sparse relationships between users and the co-occurrence of users and items in documents. For example, text documents contain words and some of these words refer to names of users. Users who co-occur together in the same documents are assumed to be socially related and their social relations are modeled by sparse Blockmodel [78]. In our case, users are the documents who adopt items modeled as words in LDA. Users do not adopt other users and items do not have any link between them.

Rosen-Zvi et al. proposed the Author Topic Model [84] to discover the topic distribution of authors for a document. However, it assumes each word in a document comes only from one author, who independently generates topics without any dependency on another author.

We extend LDA to include the notion of social correlation between users. The social correlation is calculated based on the users' and their friends' topic distributions, conditioned on the set of items adopted by the users.

Matrix Factorization

Although Matrix Factorization (MF) has not been widely applied for modeling users' items adoptions, MF has been largely successful in modeling users' items ratings. Similar to LDA, MF represents the preference of users and the characteristic of the items using low rank vectors, in real space. The general model of matrix factorization (MF) is given as follows, suppose we have an observe user item matrix represented by $Y \in \mathbf{R}^{M \times N}$, where M represents the number of items and N represents the number of users. MF derives $C \in \mathbf{R}^{M \times K}$ and $X \in \mathbf{R}^{K \times N}$, where K is significantly smaller than M or N. Each row in C represents the item's latent factor while each column in X represents the user's latent factor. To obtain C and X, MF seeks to find the matrices by

minimizing the following least squares error,

$$\sum_{m,n} (y_{m,n} - c'_m \cdot x_n)^2$$

Salakhutdinov and Mnih introduced the first probabilistic matrix factorization (PMF) with Gaussian observation noise [89], and later extended it to Bayesian Probablistic Matrix Factorization (BPMF) by providing a full Bayesian treatment through the Markov Chain Monte Carlo (MCMC) algorithm [90]. These methods produce good predictive accuracy and can scale up to large/sparse static data. The additional feature in PMF is to addition of a Gaussian noise,

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

Such that the probabilitic of observing $y_{m,n}$ is given by,

$$p(y_{m,n}|C, X, \epsilon) = \mathcal{N}\left(c'_m \cdot x_n, \sigma^2\right)$$

PMF obtains the parameters by maximizing the log likelihood using stochastic gradient descent while BPMF adds additional Gaussian and Wishart priors to use the Gibbs Sampling approach.

One well-known MF approach is Non-negative Matrix Factorization (NMF) [62, 63] which have been proposed to model image pixels and encoding variables, as well as documents and words. Based on our literature survey, we found that NMF has not been used for adoption recommendation where a user can adopt items with quantities at different points in time. The algorithm for obtaining NMF latent factors is similar to the MF with the addition of non-negativity constraints on the parameters. Chapter 5 [30] explores the use of log-barrier approach for non-negativity. An advantage of using NMF is the ease of interpretation since all the parameters are non-negative. NMF could be used as a substitute for topic models when LDA is not appropriate for certain

kinds of data representation.

Similar to Balasubramanya and Cohen [9], Ma et al. [73, 74] has also proposed a Matrix Factorization based model that jointly models the users' items ratings and the users' social network. The model from Ma et al. [73, 74] extended from the Bayesian Probabilistic Matrix Factorization (BPMF) by adding latent social factors. They jointly model two matrix factorization problems, 1) the user item rating values and 2) the user to user social links while adding constraints that the two factorization problems share similar user latent factors. Instead of defining a latent factor model for rating prediction, we model item adoption. We also avoid the additional complexity of modeling the second factorization problem by directly inferring the social correlation through the item adoptions only. Our social correlation represents how closely related the items adoptions of two users are, while Ma et al. derives a weight to model the probability of a social link between two users.

2.3 Dynamic Latent Factor Models

To derive the temporal behavioral representation of users, we utilise temporal dimension reduction to obtain low ranked vectors for the users' item adoptions. In both Chapters 4 [28] and 5 [30], we propose our variants of temporal dimension reductions by extending LDA and MF. Then in Chapter 6, we merge the proposed models from Chapters 4 and 5. Therefore, it would be necessary for us to go through some prior work in temporal models to highlight the differences in our proposed models and the contributions we make to the academic community.

There are two forms of Dynamic Latent Factor Models. The first form seeks to obtain more accurate latent factors in the temporal domain by obtaining latent factors that globally approximates the observed data [3, 15, 56, 57, 99, 109, 115]. The other form known as *online learning* focuses on the efficiency of

handling real time streaming data by maximizing the likelihood of the latent factors to fit the observed data from the most recent time window only [2, 21, 22, 41, 48, 75, 88, 107]. We are concerned with accurate latent factor representations and so we will not elaborate further about online learning, instead, we will give an overview of the various existing dynamic models in the literature.

Blei et al. proposed Dynamic Topic Model (DTM) [15] for text documents. DTM was extended from LDA to model the evolution of words within topics, i.e. The words which are prominently used in a particular topic at a particular time step will be replaced by a different set of words at a later time. However, our requirement is slightly different. We are not concern with the evolution of topic word distributions, instead, we are concern with the evolution of authors' topic distribution. The evolution of authors' topic distribution has not been considered before using LDA because LDA is mainly used for modeling text documents. Unlike human users, text documents remain static over time, so DTM which was extended from LDA does not consider the evolution of users' behavior in the way we do. When we apply LDA for modeling users' behavior, the users replace the role of documents which adopted items in place of words. In our case, we assume that topic item distributions remain static over time while the human users' evolve their preferences over time. Since the generative process in DTM does not meet our requirements, we are therefore motivated to propose our variants of Dynamic Latent Factor Models based on LDA and MF.

For modeling users' behavior, Ahmed et al. [3] used an exponential decay function to model the decay of users' search intent on search engines. But they assume that the parameters of the decay function remain constant for all topics and all users. On the contrary, we assume that there is a decay parameter for each topic and that the decay parameters vary for each user. We aim to find a way in order to estimate the decay parameter automatically and representative

of the users' temporal behavior.

To automatically decide the natural decay of each topic, Wang and McCallum [109] have proposed a non-Markovian approach to model the trend of topics evolution. Wang and McCallum approach is to associate additional Beta distribution to each topic in order to generate the time stamps of the words sampled from the topics. But this approach assumes that each topic is only relevant for each specific time period and does not directly model the evolution of users' behavior.

Since latent factors are also widely used in the collaborative filtering domain, several authors have proposed dynamic latent factor models for handling temporal data [56, 57, 99, 115]. The collaborative filtering research has always been concerned with predicting users' items ratings so their models cannot be directly applied for modeling users' items adoptions. However, due to similarity in the fundamental concept of dynamicity in latent factors, we give an overview of these work here.

Koren [56, 57] developed TimeSVD++ to address temporal dynamics through a specific parameterization with factors drifting from a central time. Koren assumed that users' items ratings remain static over time since users do not rate the same items in different time periods. However, in users' items adoptions scenario, users could adopt the same items at different time periods with different frequency.

Xiong et al. [115] extends the static case of users' items ratings from a $\mathbf{R}^{M \times N}$ matrix to a $\mathbf{R}^{M \times N \times T}$ tensor where N represents number of users, M represents number of items and T represents number of time steps. Then three set of latent factors are derived from tensor factorization as opposed to only two set of latent factors in the static matrix factorization case. The additional set of latent factor is known as the time latent factor and can be used to derive the temporal users' and items' latent factors from its multiplication. But the time latent factor in tensor factorization assumes that the items' latent factor

evolves in the same way as the users' latent factors. However, we require the items' latent factor to remain static over time while only allowing the users' latent factors to change.

Sun et al. [99] proposed Dynamic Matrix Factorization (DMF) which uses Linear Dynamical System (LDS) [40, 86, 93, 95]. The centerpiece of this work is a dynamic state-space model that builds upon probabilistic matrix factorization in [89, 90] and Kalman filter/smoothing [49, 82] in order to provide recommendations in the presence of process and measurement noises. Although the LDS component of DMF is able to model the evolution of users' behavior, the latent factors obtained by Sun et al. [99] is unbounded in the negative domain, and so will not be able to provide intuitive interpretation on the preferences of users' adoption behavior.

All of these prior work fails to satisfy the following requirements that are necessary for us to derive the temporal social correlation measures:

1. They do not explicitly model the users' items adoption data.

2. They do not obtain non-negative latent factors for easy interpretation of the users' behavior.

3. They either do not assume that users' behavior can decay over time or does not show how the users' behavior can evolve over time.

We are therefore motivated to propose our own temporal models during the research process of this dissertation. We first proposed our own Decay Topic Model in Chapter 4 to model users' decay and obtain preliminary results to validate our temporal social correlation. We then combine the use of Non-negative Matrix Factorization (NMF) for parameter estimation and proposed different variants of DMF for different adoption scenarios in Chapter 5. We show that Linear Dynamical Systems (LDS) which was extended to Dynamic Matrix Factorization (DMF) by Sun et al. [99], has parameters that can first be solved by NMF to satisfy the non-negativity constraints. We use NMF

for obtaining the item latent factor matrix and initial values of the users' latent factors. Then we apply Kalman filtering and RTS smoothing for each individual user to obtain better estimates of their latent factors. Finally, we merge LDA and LDS to obtain the Linear Dynamical Topic Model (LDTM) in Chapter 6. LDTM is able to model the users' item adoption data, obtain probabilistic (non-negative) latent factors for characterizing users' behavior over time and automatically infer the optimal decay parameters.

2.4 Information Cascades and Diffusion of Innovation

There are alternative concepts of Influence that is different from what we described in Section 2.1. These prior research loosely relate the concepts of information cascades, propagation and diffusion of innovation with influence without identifying the homophily and selection effects.

Kempe et al. [51] proposed two diffusion models which led to their use in many other influence work [24, 25, 42, 43]. The two models are known as the Independent Cascade Model (ICM) and the Linear Threshold Model (LTM). In these models, a user is said to be diffused if they have adopted the item after being influenced by their neighbors. ICM assumes that diffusion can take place from a user to another user if there exists a path of diffused users between them. Whether a user decides to adopt depends on LTM, which states that every user has a threshold of adoption and have a certain amount of pre-defined influence probability on other users. A user adopts when the influence from their neighbor exceeds some threshold.

To demonstrate the application of ICM and LTM, Kempe et al. [51] study the problem of *Influence Maximization*, which is to find an initial set of adopters who can propagate the message throughout the network to maximize the number of adopters after a certain duration of time.

Chen et al. [24, 25] subsequently extended the influence maximization problem by proposing improved algorithms to find the initial set of seed users. But [24, 25, 51] assumed that the probability of influence between users are a pre-defined value specify using the basis of prior knowledge. Goyal et al. [43] then proposed the estimation of probabilities between users call this the General Threshold Model.

Instead of inferring for the influence probabilities, Gomez-Rodriguez et al. [42] extends the Independent Cascade Model to address the problem of inferring the hidden edges in a directed social network based on the observation of infected nodes in the network. Cosley et al. [31] extended ICM and LTM to include temporal information and call the new measure as K-exposure to estimate influence and showed that such influence also exists in Wikipedia communities.

In a related domain which studies diffusion of innovation, Bass proposed a diffusion model at the macro perspective for predicting the number of adopters for an item [10]. The model assumes two kinds of users in the market, imitators and innovators. Innovators adopt items independently and influence the imitators to adopt the items. But the Bass model predicts item adoptions at a macro and aggregated level across all users without quantifying the micro interactions for every pair of users. Luu et al. [72] extended Bass Model to take into account of an exponential model. Yang and Leskovec [117] also model diffusion for an implicit network by using a nonparametric diffusion model.

2.5 Other Influence

Another simple way of measuring influence is to count the amount of text that has been duplicated [97], number of retweets in Twitter [23, 83, 118], or the diffusion of URL by the original user who posted it [8, 23].

Snowsill et al. [97] defined influence in a network when a user copies the

text from another user for producing their content. Snowsill et al. uses a hierarchical tree structure to represent text, then extended the NetCover algorithm for measuring the amount of duplication.

Cha et al. [23] defined influence of a user in Twitter network as an aggregated measure of the number of followers a user has, the number of times the user is retweeted and the number of mentions in the tweets of other users. Romero et al. [83] extends the HITS algorithm [54] to model influence and passivity based on retweets. Zhang et al. [118] uses the retweeting behavior as a measure of influence to determine whether locality plays a role in the amount of influence.

Bakshy et al. [8] studied the effectivess of diffusion on the Twitter network and the costs involve in employing users help in disseminating a message. Bakshy et al. studies conclude that every user on the network is able to diffuse the message to a certain extent and it may not be the most cost effective in seeking only the prominent diffusers on the network. Companies will do better in their marketing campaign in finding a large initial set of average diffusers to spread the message.

However, these prior works consider the cascade or diffusion of items in an explicit manner. As a result, they could either handle a single item during the cascading process or require high computation costs for a very large number of items during diffusion. To model a set of items for diffusion or inferring influence, many authors [33, 36, 39, 70, 71, 100, 101, 104, 110, 111]. have turned to the use of Topic Models for dimension reduction.

Cui et al. proposes an influence matrix to suggest what items a user should share to maximize their individual influence in their own community [33]. Their matrix measures influence between users and items while ours measure between users and users.

Gerrish and Blei extended the dynamic LDA to identify the most influential documents in a scientific corpus [39]. But dynamic LDA assume that the

documents' latent factors evolve only with small perturbations while words' latent factors evolve over time. Gerrish and Blei's work differ from ours because we allow greater variability in users' (documents') latent factors and assume that items' (words') latent factors remain constant.

Similar to Romero et al. [83], Weng et al. [111] extended Pagerank [19] to include topic models in the computation of influence between users. Apart from [39, 104], all the prior works which uses dimension reduction does not use the time information when inferring influence. In that aspect, our dissertation goes beyond the norms by considering temporal users' items adoption and proposing several temporal models.

A notable challenge to what we do is the work proposed by Ver Steeg and Galstyan [104]. Ver Steeg and Galstyan also use Topic models to reduce the dimensionaity of item adoptions follow by an information theoretic measure of causality, known as *Transfer Entropy*, for measuring social influence. An expression[1] for measuring Transfer Entropy between two time series of i and j is given by

$$TF(j \to i) = I(i_t; j_{t-1}|i_{t-1}) = H(i_t|i_{t-1}) - H(i_t|i_{t-1}, j_{t-1})$$

where H is entropy, a measure of uncertainty. When $TF(j \to i) > 0$ it implies that j_{t-1} reduces our uncertainty about i_t. The reduction in uncertainty can be seen as an additional feature which we could use in a statistical model to make future predictions of i. As a result, Transfer Entropy is used as a measure of influence in [103, 104]. The algorithm to estimate Transfer Entropy is based on the nearest neighbor approach developed in statistical physics [58, 59, 105].

But there are significant drawbacks to Ver Steeg and Galstyan [104]. This approach makes no assumption on the joint distributions of the variables, so it requires many time steps for achieving accurate estimation. It also ignores

[1]Transfer Entropy is also known as Conditional Mutual Information $I(X;Y|Z)$ and is defined as $\int_{z\in Z} p(z) D_{KL}[p(X,Y|z)||p(X|z)p(Y|z)]$. The expression we show is derived from the definition and provide insights for its application as a causality measure.

the temporal correlations between users' topic distributions and the users' behavior evolution. We distinguish our work in Chapter 6 by proposing a Linear Dynamical System (LDS) approach to linearly correlate the users' topic distributions, and using Granger causality that likewise assumes linear relationship among variables. Due to the additional linear assumption imposed by the Granger causal measures, it requires less number of time steps for us to derive an accurate measure of social correlation between the users.

Chapter 3

Social Correlation for Static Data

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).

3.1 Objectives

In this chapter, we aim to address 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 [55]. One such model is Latent Dirichlet Allocation (LDA) [16], 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.

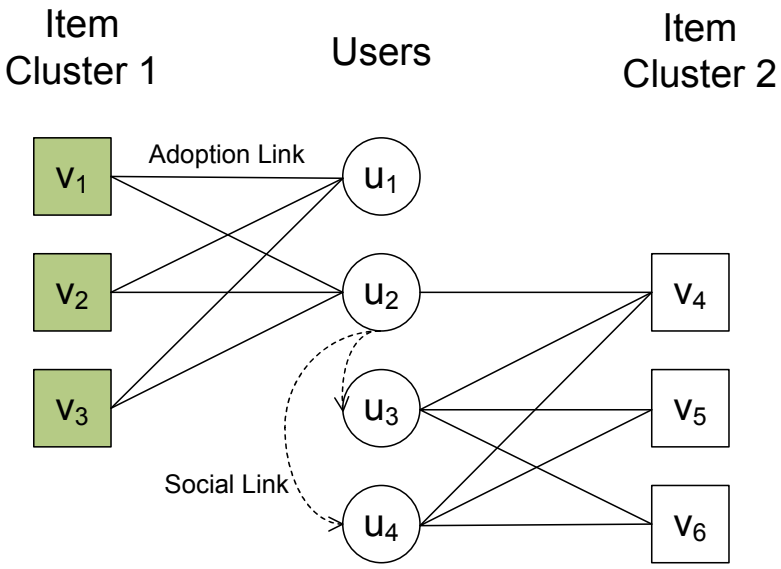


Figure 3.1: Example Scenario of Adoption (solid) and Social Links (dotted)

Consider the example scenario in Figure 3.1. 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 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.

This chapter is organized as follows:

1. We incorporate the social correlation matrix C in the generation of user-item adoption links. 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 principled, and unified way. The 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 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.

The rest of the chapter is organized as follows. We establish the existence of correlation between adoption and social links in Section 3.2 through hypoth-

esis testing. In Section 3.3, we introduce the social correlation measure that is derived from the users' latent factors and their item adoptions. In Sections 3.4 and 3.5, we describe two generative models: Sequential and Unified respectively, and show how their parameters can be learned efficiently. We then proceed to evaluate our methods in Section 3.6. Finally we end this chapter in Section 3.7.

3.2 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 [96] 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 3.1. 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 3.1).

Table 3.1: 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

Table 3.2: 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 (131,126,176)	2,485,417 (2,640,636)	133,766,812
Has Social Link	150,316 (305,535)	161,372 (6,153)	311,688
Total	131,431,711	2,646,789	134,078,500

Table 3.3: 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 (80,103,462)	2,874,403 (2,893,831)	82,997,293
Has Social Link	112,575 (132,003)	24,197 (4,769)	136,772
Total	80,235,465	2,898,600	83,134,065

We perform hypothesis testing using the Fisher Exact Test [37]. 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 Tables 3.2 and 3.3. 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.

3.3 Social Correlation Measure

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

$$E \approx \Phi \cdot \Theta \cdot C^T \tag{3.1}$$

where Φ represents the latent factors of items arranged in a $|V| \times |Z|$ matrix with Z being the set of latent factors, Θ 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 minimzing 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.

3.3.1 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.

3.3.2 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 Θ and the latent factors of items Φ.

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 and the user herself. We may restate the equation as follows,

$$\begin{aligned} P(e_{v,u}|\Theta,\Phi,F) &= \sum_{x\in F_u} P(e_{v,x}, f_{u,x}|\Theta,\Phi,F) \\ &= \sum_{x\in F_u} P(e_{v,x}|\Theta,\Phi)P(f_{u,x}|F) \end{aligned} \tag{3.2}$$

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 3.2 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:

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

where F the social network is always available. The information to be learnt are Θ, Φ and C.

3.3.3 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 3.1) 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.

3.4 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 θ and ϕ

variables in the graphical model as shown in Figure 3.2. We also assume the existence of a social network as reflected by the shaded f variables.

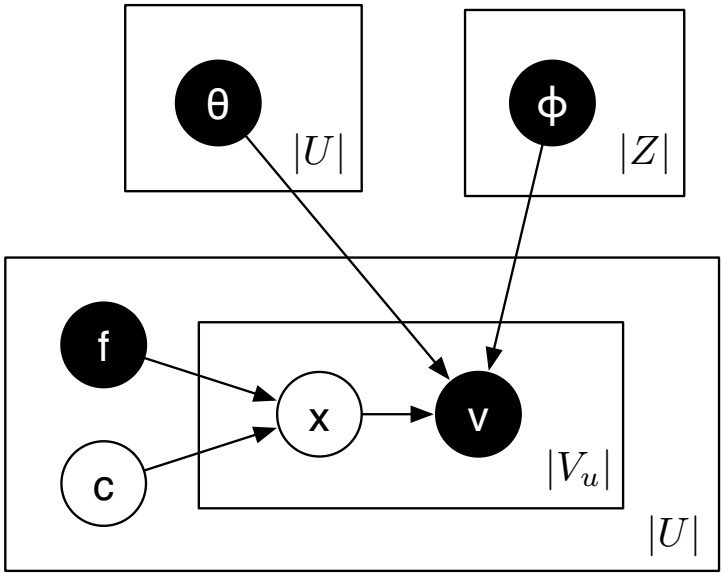


Figure 3.2: Sequential Generative Model for Static Social Correlation

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 the *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 Θ and Φ 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 [35],

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

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.

3.4.1 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,

$$\begin{aligned} P(x_{v,u} = x | e_{v,u}, \Theta, \Phi, F, C) &= \frac{P(e_{v,x}|\Theta,\Phi)P(x|C_u, F_u)}{\sum_{x' \in F_u} P(e_{v,x'}|\Theta,\Phi)P(x'|C_u,F_u)} \\ &= \frac{e'_{v,x}c_{u,x}}{\sum_{x' \in F_u} e'_{v,x'}c_{u,x'}} \\ &= h(u,x,v) \end{aligned} \tag{3.3}$$

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

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

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 λ_u [18] and proceed to solve the following using differentiation,

$$\begin{aligned} \frac{d}{d\ c_{u,x}} \left[\sum_{v \in V_u} \sum_{u \in U} \log \Big(\sum_{x \in F_u} e'_{v,x} c_{u,x} \Big) - \lambda_u \Big(\sum_{x \in F_u} c_{u,x} - 1 \Big) \right] &= 0 \\ \sum_{v \in V_u} \frac{e'_{v,x}}{\sum_{x' \in F_u} e'_{v,x'} c_{u,x'}} - \lambda_u &= 0 \\ \lambda_u = \sum_{v \in V_u} \frac{e'_{v,x}}{\sum_{x' \in F_u} e'_{v,x'} c_{u,x'}} \\ \lambda_u c_{u,x} = \sum_{v \in V_u} \frac{e'_{v,x} c_{u,x}}{\sum_{x' \in F_u} e'_{v,x'} c_{u,x'}} \\ c_{u,x} = \frac{1}{\lambda_u} \sum_{v \in V_u} \frac{e'_{v,x} c_{u,x}}{\sum_{x' \in F_u} e'_{v,x'} c_{u,x'}} \end{aligned} \tag{3.4}$$

Recall in our E step that we have calculated something similar to the RHS of Equation 3.4. By inserting the results of Equation 3.3 from the E Step, we

get

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

where λ_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.

3.4.2 Complexity Analysis

In Section 3.2, 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, $|F| << |U|^2$ and $|E| << |V| \cdot |U|$. 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 3.3 of the E Step and Equation 3.4 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, $O(|U| \cdot avg(|V_u|) \cdot avg(|F_u|))$. Expressing in terms of number of edges,

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

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 3.6.5.

3.5 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: Θ, Φ and C. The sequential approach we took in Section 3.4, 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 Φ, Θ and C.

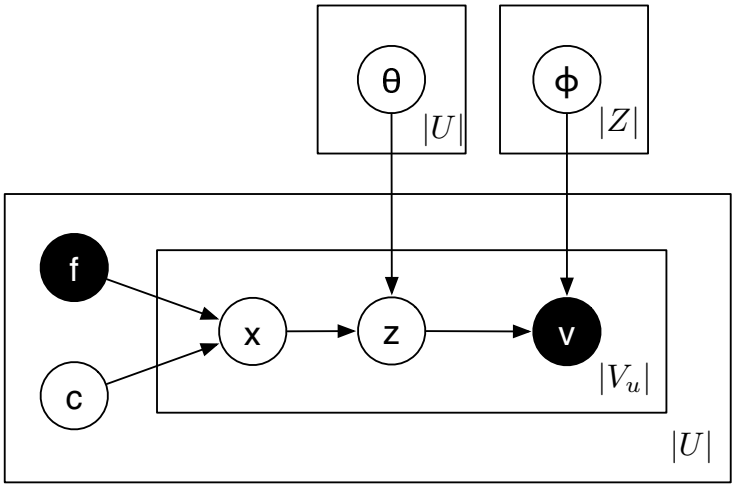


Figure 3.3: Unified Generative Model for Static Social Correlation

Figure 3.3 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 θ_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 ϕ_z, u chooses an item v to adopt.

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

3.5.1 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.1.

The E Steps are

$$f(u, v, z) = \frac{\phi_{z,v}\theta_{u,z}c_{u,u}}{\sum_{z' \in Z}\sum_{x' \in F_u}\phi_{z',v}\theta_{x',z'}c_{u,x'}} \tag{3.5}$$

$$g(u, v, z) = \frac{\sum_{x \in F_u}\phi_{z,v}\theta_{x,z}c_{u,x}}{\sum_{z' \in Z}\sum_{x' \in F_u}\phi_{z',v}\theta_{x',z'}c_{u,x'}} \tag{3.6}$$

$$h(u, v, x) = \frac{\sum_{z \in Z}\phi_{z,v}\theta_{x,z}c_{u,x}}{\sum_{z' \in Z}\sum_{x' \in F_u}\phi_{z',v}\theta_{x',z'}c_{u,x'}} \tag{3.7}$$

The M Steps are,

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

3.5.2 Complexity Analysis

As mentioned in Section 3.4.2, 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 3.5, 3.6 and 3.7. For each user u, Equations 3.5, 3.6 and 3.7 requires $O(|Z| \cdot |V_u| \cdot |F_u|)$. So the complexity for all users is given by $O(|U| \cdot |Z| \cdot avg(|V_u|) \cdot avg(|F_u|))$. Following the previous analysis on the complexity, The complexity is given by,

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

3.6 Experimental Evaluation

3.6.1 Experimental Setup

Data Set: For experiments, we extracted data sets from the LiveJournal data set and Epinions data set described in Section 3.2. 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 3.4 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 3.1 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 3.4: 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 3.4 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 2.2 (*LDA*), Section 3.4 (*Sequential*), and Section 3.5 (*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.

3.6.2 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 3.4, 3.5 and 3.6 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.

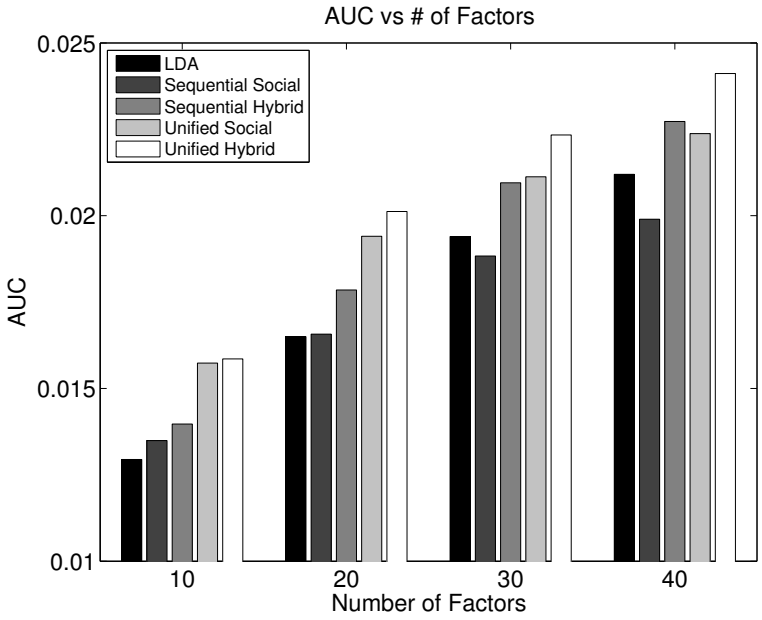


Figure 3.4: LiveJournal: AUC vs Number of Factors

In Figures 3.5 and 3.6, we show that our performance is consistent across all latent factors. So 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 A.2 shows the list of top ranked items for a subset of the topics. The items in these topics give us a qualitative view of whether the chosen number of topics is appropriate.

3.6.3 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. Given that we have the

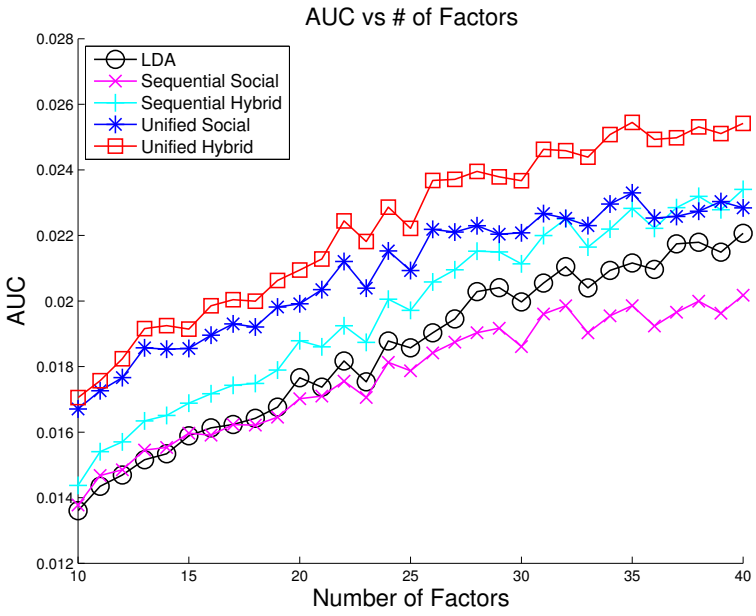


Figure 3.5: LiveJournal: AUC vs Number of Factors

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 equal-width 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 $c_{u,u} \in (\frac{2}{3}, 1]$. 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 3.7 and 3.8 show the results of LiveJournal and Epinions for the mean ratios using the *Sequential Model*. In each figure, a higher bar indicates a

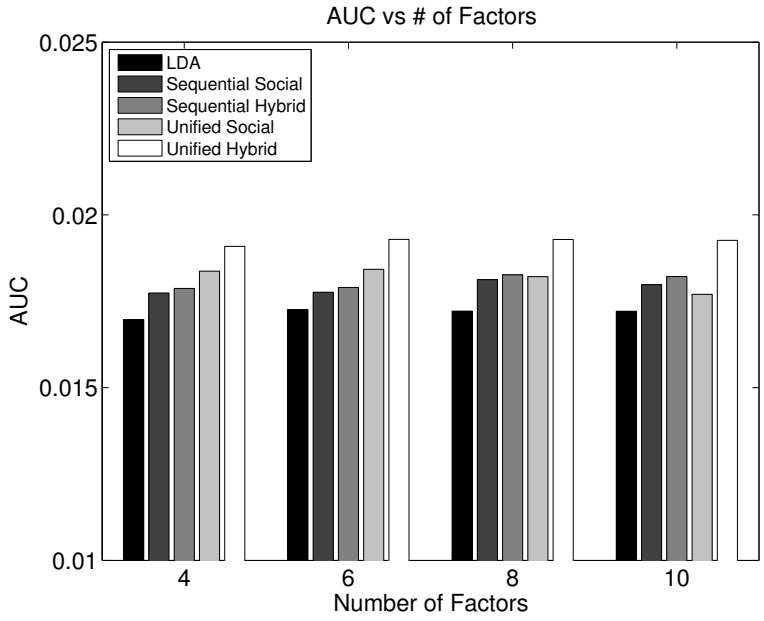


Figure 3.6: Epinions: AUC vs Number of Factors

better performance over the baseline method Random. AUC ratio ≈ 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.

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 3.9 and 3.10 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

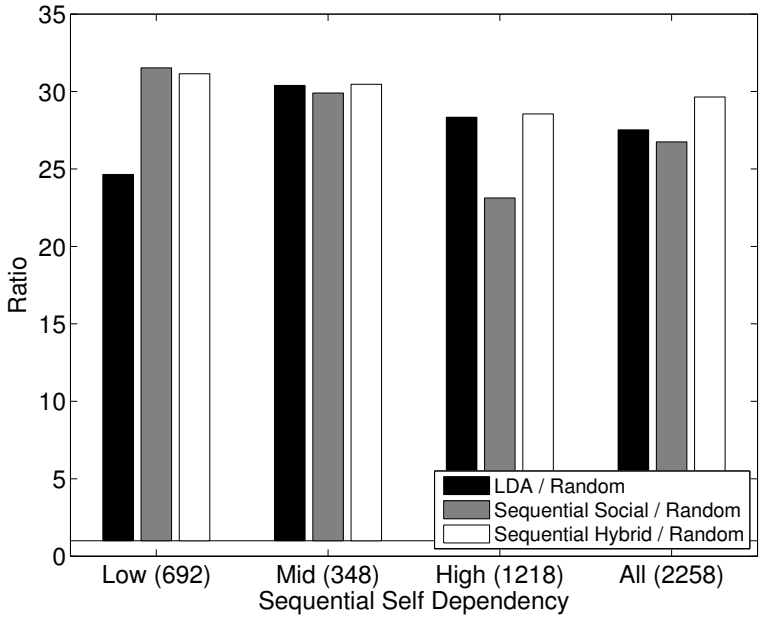


Figure 3.7: LiveJournal: Sequential Model AUC Ratio vs Self-Dependency

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 Model*s 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 A.3 for further analysis on the self-dependency values.

3.6.4 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 3.11 and 3.12 show the AUC ratio with respect to the log of the number of items (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

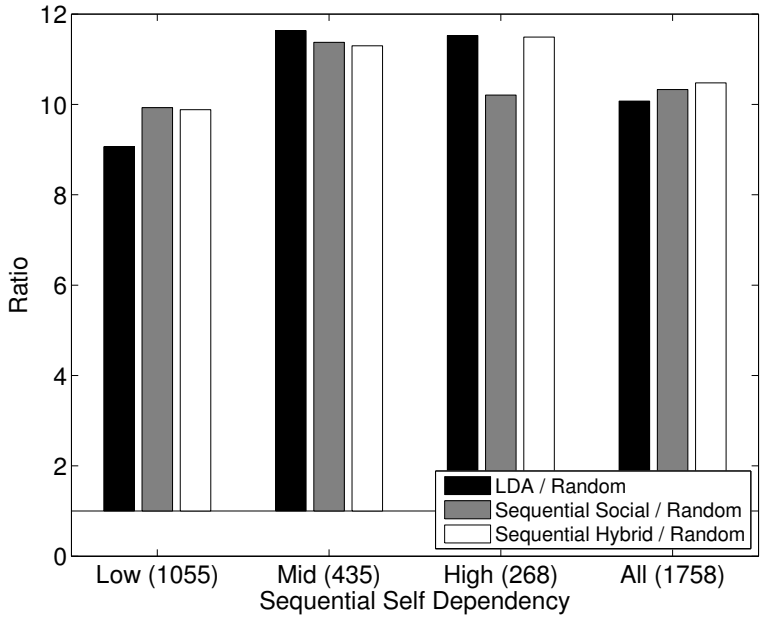


Figure 3.8: Epinions: Sequential Model AUC Ratio vs Self-Dependency

parallel to the y-axis gives the median value for the number of items each user has. As shown in Figure 3.11 for LiveJournal, Social outperforms LDA for approximately half of the users. For Figure 3.12 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 3.11 and 3.12 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. A theoretical estimate of Random's AUC with respect to the number of items can be found in A.4.

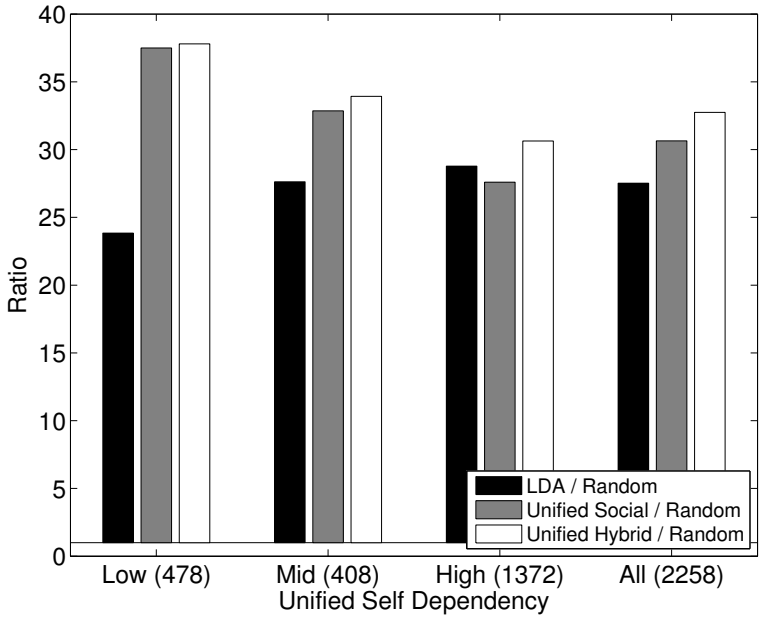


Figure 3.9: LiveJournal: Unified Model AUC Ratio vs Self-Dependency

3.6.5 Convergence Rate

We explained the complexity of the algorithm in Section 3.4.2 and Section 3.5.2. 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 3.13 and 3.14 show the likelihood with respect to number of iterations for LiveJournal and Epinions respectively. Since we have pre-computed *LDA* for the *Sequential Model*, the likelihood given by *LDA* is therefore a constant as shown by the red line in Figures 3.13 and 3.14. 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 Epin-

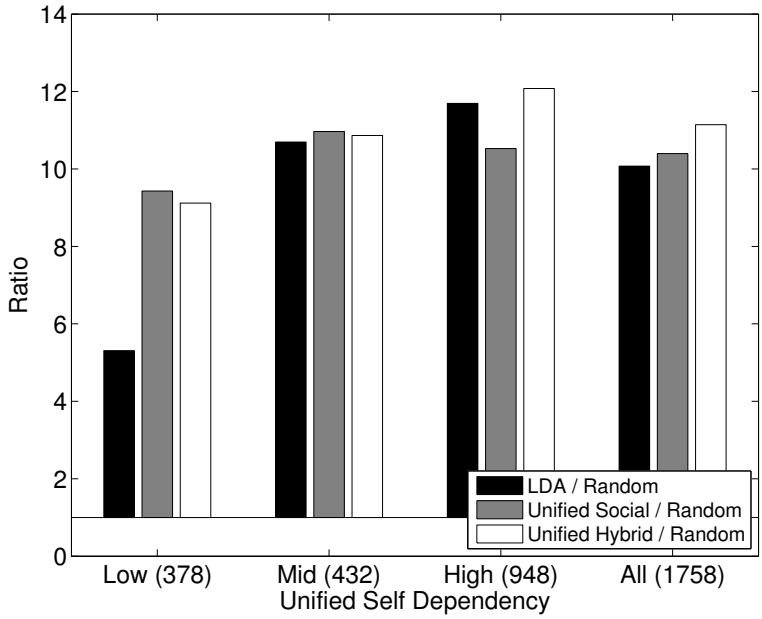


Figure 3.10: Epinions: Unified Model AUC Ratio vs Self-Dependency

ions, 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.

3.6.6 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). 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 3.15 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

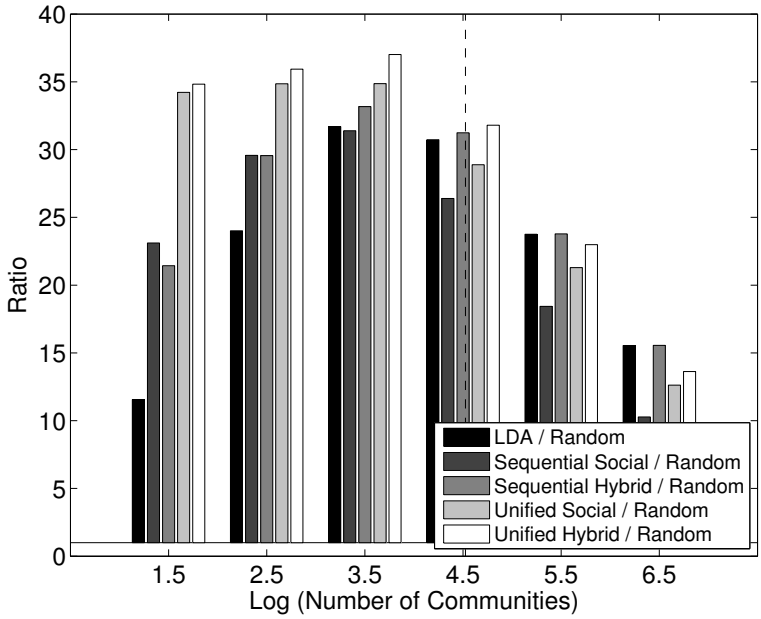


Figure 3.11: LiveJournal: AUC Ratio vs Log (# Communities)

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 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 3.16 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*

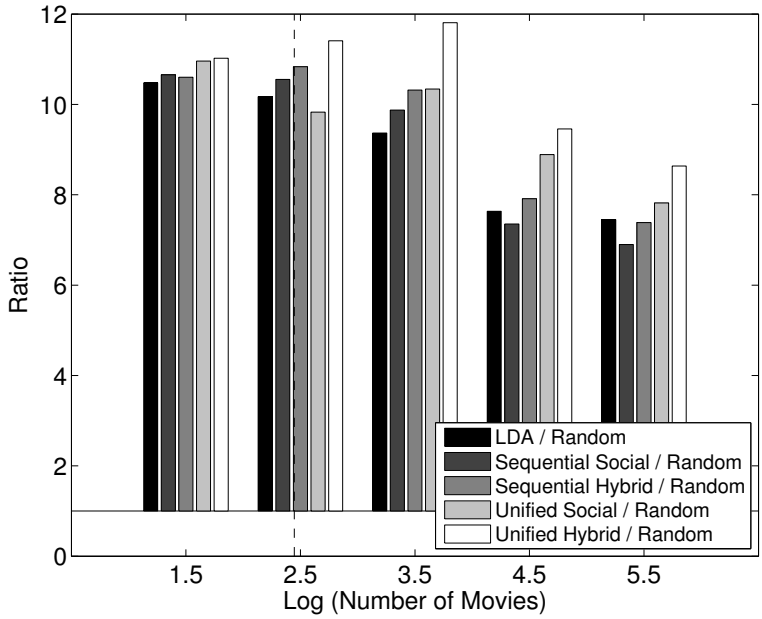


Figure 3.12: Epinions: AUC Ratio vs Log (# Movies)

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.

3.7 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 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

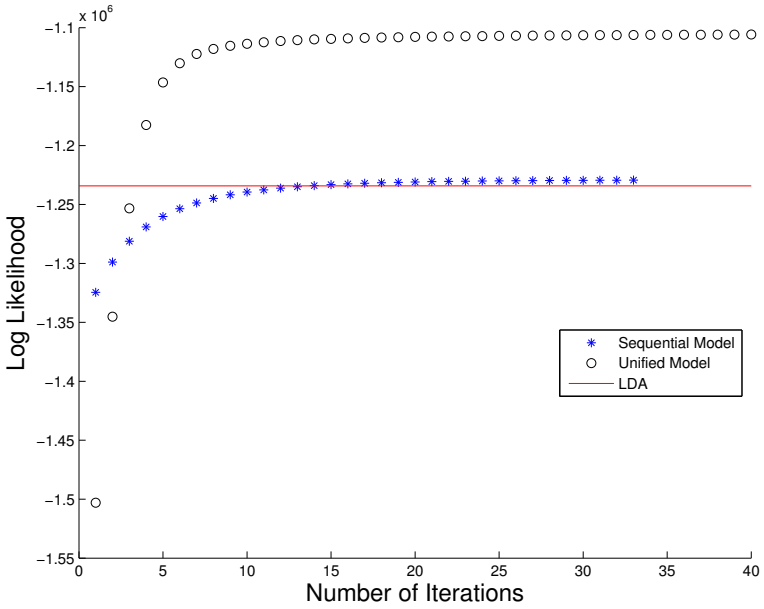


Figure 3.13: LiveJournal: Log Likelihood vs Number of Iterations

outperform the approach based on latent factors only (LDA).

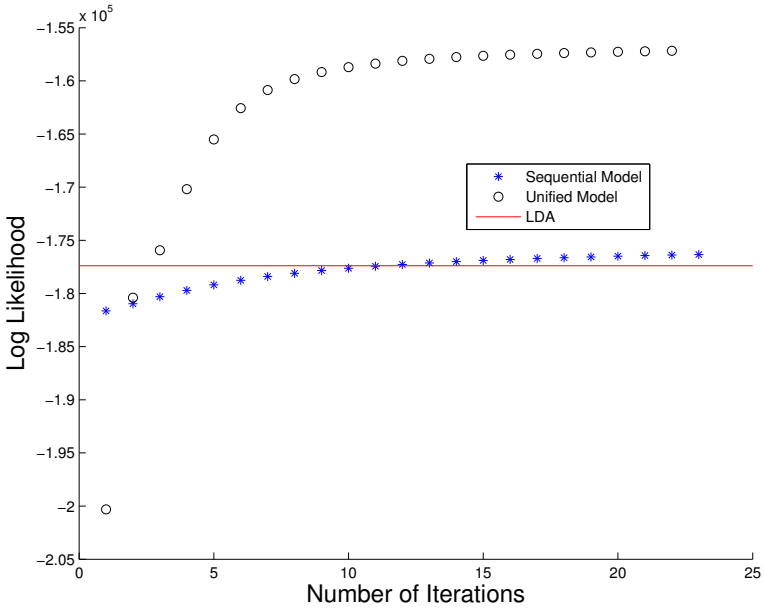


Figure 3.14: Epinions: Log Likelihood vs Number of Iterations

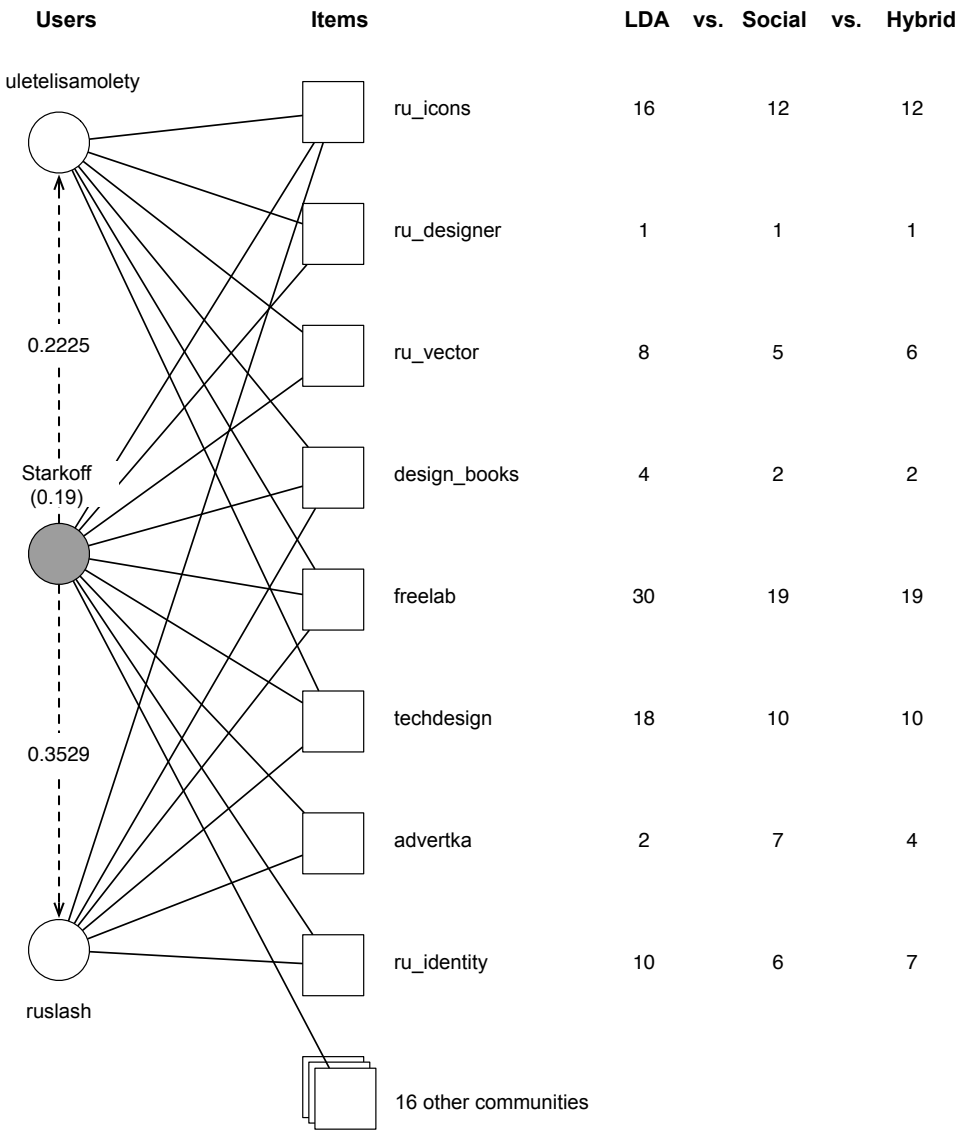


Figure 3.15: LiveJournal: Low Self Dependency

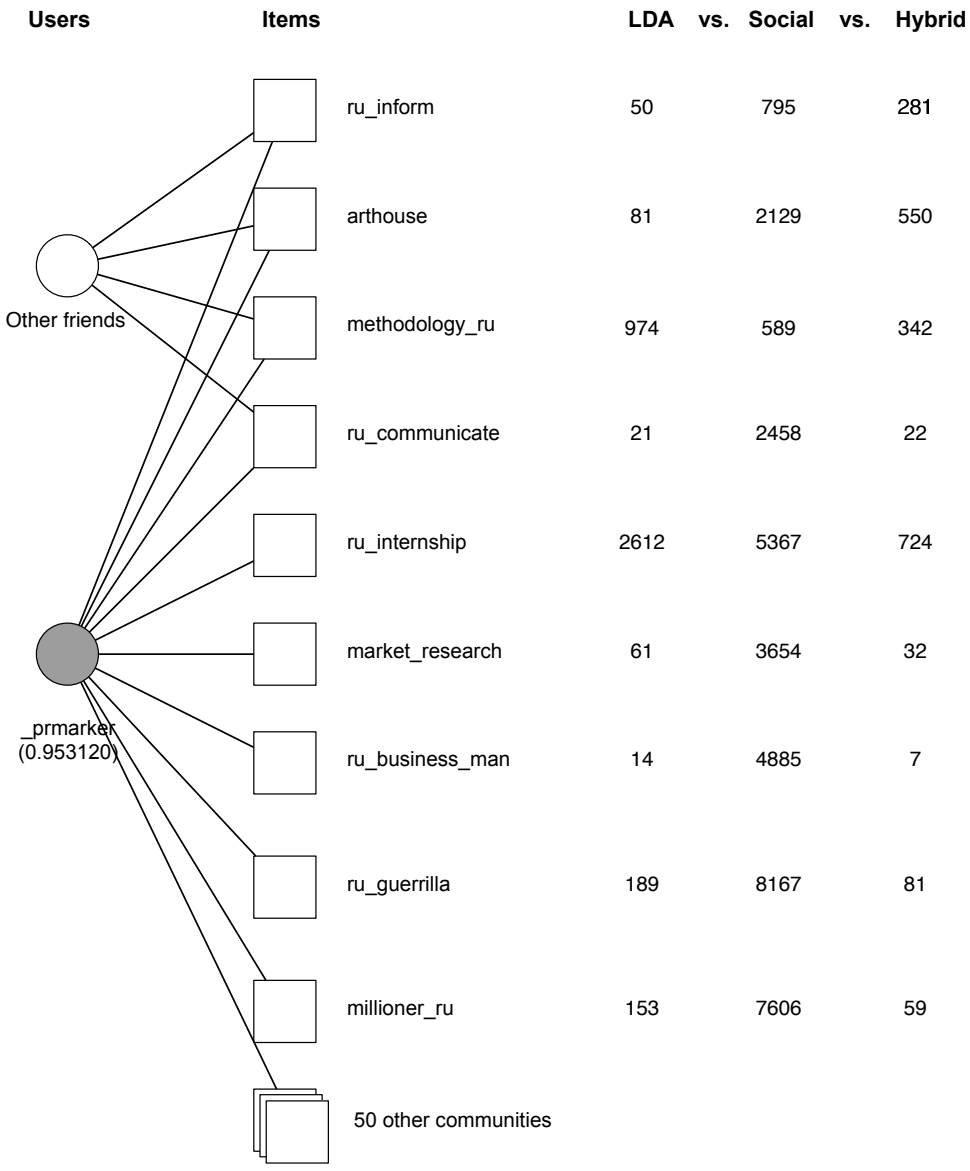


Figure 3.16: LiveJournal: High Self Dependency

Chapter 4

Decay Topic Model and Two-period Temporal Social Correlation

User-to-user interactions have become ubiquitous in Web 2.0. Users exchange emails, post on newsgroups, tag web pages, co-author papers, etc. Through these interactions, users co-produce or co-adopt content items (e.g., words in emails, tags in social bookmarking sites). We model such dynamic interactions as a user interaction network, which relates users, interactions, and content items over time. After some interactions, a user may adopt content that is more similar to those adopted by other users previously. We term this effect *temporal social correlation*, and we seek to mine from such networks the degree to which a user may be socially correlated with another user over time. We propose a *Decay Topic Model* to model the evolution of a user's preferences for content items at the topic level, as well as a *Temporal Social Correlation Measure* that quantifies the extent of temporal social correlation based on interactions and content changes.

4.1 Motivation

User interactions in a dynamic social network provide insights for the evolution of relationships among a set of users. The user interactions in this dynamic social network lead to the *production* or *adoption* of content items covering a set of evolving latent factors. Using these evolving latent factors, we aim to derive the temporal relationships among the users. We define **Two-period Temporal Social Correlation** as a temporal correlation between (a) the latent factors in the current time step of the target user, and (b) the latent factors in the previous time step of other users she interacts with. The degree of correlation capture the extent to which the target user depends on the other users, which explains the change in her latent factors. For brevity, we will use *Temporal Social Correlation* in this chapter instead of *Two-period Temporal Social Correlation*.

User Interaction Network: A user interaction network consists of interactions that adopt new content items over time. We consider a general approach of defining an interaction d (e.g., an email exchange, a published paper) as a tuple $\langle A_d, W_d, \tau_d \rangle$ where A_d, W_d and τ_d denote the set of users, content items (e.g., words in an email or paper), and time point of the interaction respectively. We represent a set of interactions over a time period as a graph called *user interaction network*, as shown in Figure 4.1. Users, interactions, and content items are the vertices in the user interaction network example. An edge connects a user a to an interaction d taking place at time τ, which a participates in. Similarly, we draw an edge from d to each content item w adopted through d. This network has three interactions: $d_1 = \langle \{a_1, a_2\}, \{w_1, w_2\}, \tau_1 \rangle$, $d_2 = \langle \{a_1, a_3\}, \{w_3, w_4\}, \tau_2 \rangle$, and $d_3 = \langle \{a_2, a_3, a_4\}, \{w_1, w_2\}, \tau_3 \rangle$.

The user interaction network or interaction network can be found in many situations involving user communication of one form or another. In an email-based user interaction network, users produce (adopt) email content as they interact with other email users by replying to email threads. In a newsgroup-

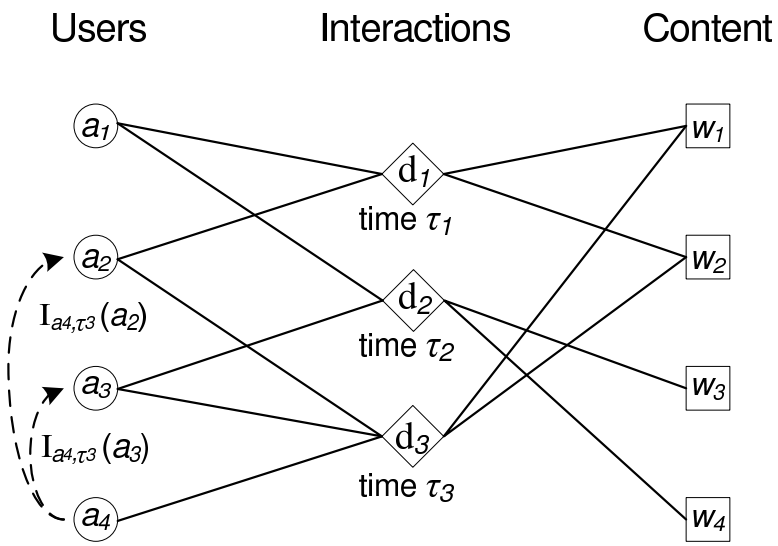


Figure 4.1: User Interaction Network

based user interaction network, users submit news posts as they respond ("interact") to other users' news posts. As Web 2.0 and social media sites become very popular, we can find even more interaction networks.

Temporal Social Correlation: From the interaction linkages among users and their evolving latent factors, one can observe the dependencies among users. An email user may change her email content after exchanging emails with another email user. Similarly, a newsgroup user may change news content in her posts after reading news posts from another user. In both cases, we say the first user is *socially dependent* on the second user if the former adopts content that is more similar to the latter after some interaction between them.

We use the scenario in Figure 4.1 to illustrate the notion of temporal social correlation. Suppose that the three interactions occur at different time points $\tau_1 < \tau_2 < \tau_3$. At τ_3, the interaction between a_2, a_3, and a_4 result in the co-adoption of content items w_1 and w_2. We are interested in whether a_4 is socially more dependent on a_2 or on a_3 for adopting the items w_1 and w_2. The dotted lines represent the temporal social correlation links, the direction implies who is dependent on whom, and the weight signifies the extent of dependency. To answer this question, it is instructive to look at the previous time points τ_1 and τ_2. It is evident that since a_2, but not a_3, has been previously associated

with w_1 and w_2 before τ_3, so it is likely that a_4 is socially dependent on a_2 for the adoption of w_1 and w_2, rather than on a_3.

Temporal social correlation is therefore defined based on two key criteria: (a) interactions between two users; and (b) content changes of the user who depends on the other user. As interactions can be ordered by time, we study precedence between interactions by considering a snapshot representation of interactions by sampling the network at different time points. From the snapshots, we derive the *set of interactions occurring at time step t* by $D_t = \{d|\tau_d \in t\}$. For a sequence of multiple time steps T, we have interactions $D_T = \bigcup_{t \in T} D_t$.

The second criteria, content change, can be modeled in different ways. A straightforward approach is to model content as a bag of words and content change is then measured by difference in word usage. This approach however does not work well as word usage can be noisy. Instead, we adopt the topic modeling approach which determines the latent factors as topics behind the observed words. Content change can therefore be measured by a change in topics.

Problem Statement: The research problem of modeling temporal social correlation is thus defined by: *Given a set of users with interactions D_T over a sequence of time steps T, determine the temporal social correlation between a_i on another user a_j at time step t, $I_{a_i,t}(a_j)$, for every $a_i, a_j \in A$ and every $t \in T$.* A is the set of all users in D_T. $I_{a_i,t}(a_j) \in [0, 1]$ such that 0 and 1 represent no dependence and complete dependence respectively. Temporal social correlation is time step specific so as to capture its evolution. The correlations may exist among users at a time step only when these users have interactions within the same time step. Otherwise, they are deem to be socially independent of one another.

Modeling temporal social correlation comes with the following research challenges.

- *Dynamic changes in topics of interaction content:* The existing topical models are designed primarily for static content. To cope with emerging new interactions and users, we need to develop new and efficient topic models that can model dynamically changing interaction content.

- *Missing user interaction data:* User interactions do not occur with the same intensity in all time steps. They may be dense in some time steps, but sparse or even missing in others. Even in the case of missing data for a given user in a time step, we still need to model how the user's topic preferences are related to those of other users.

- *Smooth transition of user topic preferences:* Users normally do not change their topical preferences abruptly. Hence, the challenge is how to model the smooth transition in user's topical preferences.

- *Temporal Social Correlation measurement:* It is expected that a user may be correlated with more than one other user, each potentially with a different quantity. Thus, we need to develop the principles in which these quantities can be derived from the interactions.

To handle these challenges, we propose the Decay Topic Model and measurement of Two-period Temporal Social Correlation. We apply temporal social correlation to the prediction of future user topic preferences on two real datasets extracted from DBLP [67] and ACM Digital Library [1]. Compared with a baseline method, our proposed prediction method using temporal social correlation derives more accurate prediction of future topic preferences.

The rest of this chapter is organized as follows. In Section 4.2, we describe the decay topic model and our measure of dependency. We then proceed to evaluate our method in Section 4.3. Finally we end the chapter with in Section 4.4.

4.2 Temporal Social Correlation

In this chapter, we are interested in modeling the evolution of user interaction networks so as to derive temporal social correlation. In particular, we observe that there are two main components in the evolution of user interaction networks, namely:

1. The change in user preferences for different content items over time.

2. The change in social correlation between users over time.

Each of these two components can be represented formally as networks induced from the original user interaction network as follows.

Content Network: This network relates users to content items that they produce or adopt through interactions. For a given set of interactions D_t occurring at time t, an edge (a, w) exists if $\exists d \in D_t,\ a \in A_d \wedge w \in W_d$. Figure 4.2 illustrates three content networks over three time steps $t_1 = \{\tau_1\}$, $t_2 = \{\tau_2\}$, $t_3 = \{\tau_3\}$, induced from the interactions in Figure 4.1.

Temporal Social Correlation Network: This network relates users to other users whom they may socially correlate with. A directed edge from a_i to a_j exists if a_i has social correlation with a_j. The edge weight $I_{a_i,t}(a_j)$ reflects the degree to which a_i is socially correlated with a_j at time step t. A loop indicates a user a_i's self-dependency with weight $I_{a_i,t}(a_i)$. In this work, we assume social correlation can be inferred from interactions. Therefore, we only draw an edge from a_i to a_j at time t, if both participate in at least one interaction at time t, i.e., $\exists d \in D_t,\ a_i, a_j \in A_d$. Figure 4.3 illustrates how the temporal social correlation network evolves over three time steps, induced from the interactions in Figure 4.1.

Given a user interaction network spanning the time period T, the problem we address here is determining the temporal social correlation measure $I_{a_i,t}(a_j)$ for every $a_i, a_j \in A$, and $t \in T$. In the following sections, we will describe how we can model users' content changes at the topic level from the

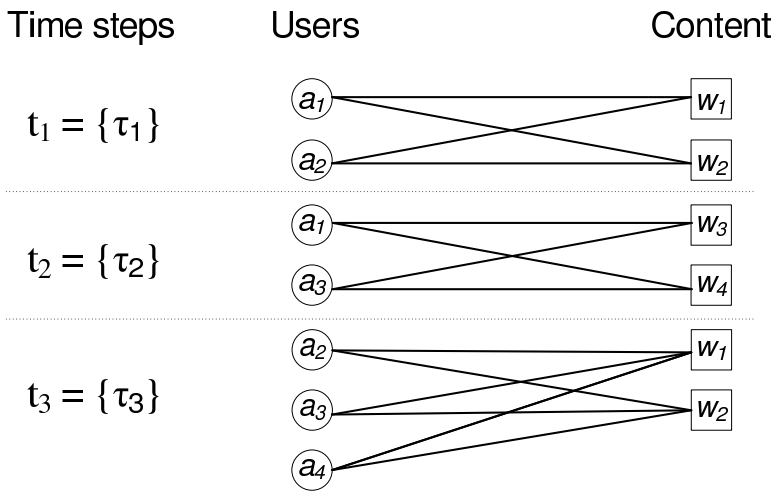


Figure 4.2: Evolving Content Network

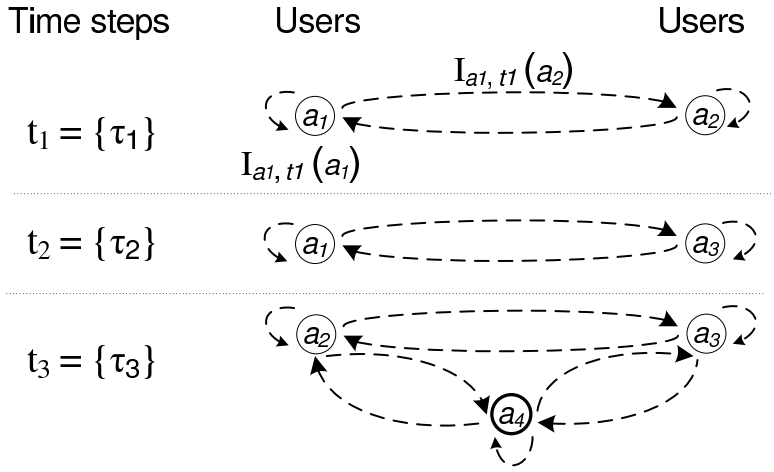


Figure 4.3: Evolving Temporal Social Correlation Network

evolving content network. We will then show how the temporal correlation of content changes between users reveals the edge weights in the temporal social correlation network over time.

4.2.1 Topic Models for Evolving Content Network

While a content network reveals the various content items adopted by a user, it may not show the user's underlying topic preferences that give rise to the adoption of those content items. The reason is that content items may be noisy. For instance, in different interactions, a user may adopt different words (e.g., "Porsche", "Ferrari") that actually refer to the same topic (e.g., luxury cars). This motivates us to model a user a's content as a topic distribution $\theta_{a,t}$

derived from the content network at time t. As the content network evolves, so does a's topic distribution, i.e., $\theta_{a,t}$ varies with t. In the following, we will first model a user's topic distribution in a static manner, before moving on to our proposed temporal-based *Decay Temporal Model*.

Static Topic Model

We observe that the bipartite structure of the content network resembles the relationship between documents and words. Just as a document contains a bag of words, a user is associated with a bag of content items from various interactions. As a naive baseline, we consider topic modeling techniques for text documents in order to model the static topic distribution of users. One such technique is Latent Dirichlet Allocation or LDA [16].

LDA can be adapted to our context as follows. To facilitate the presentation of our model, we introduce a set theoretic notation to explain the variables. Let Z denote the set of topics. For each $z \in Z$, ϕ_z denotes the topic z's item distribution. Each ϕ_z is modeled as a Dirichlet Distribution of V dimensions where V is the total number of unique content items (non-stop words) in the interaction network (corpus).

Let A denote the set of users. For each $a \in A$, θ_a denotes a's topic distribution. Each θ_a is modeled as a Dirichlet Distribution of K dimensions, where K is the number of topics in the set Z. To put it more formally, we have:

$$\phi_z \sim Dirichlet(\beta), \quad \beta \textit{ is a constant}$$
$$\theta_a \sim Dirichlet(\alpha), \quad \alpha \textit{ is a constant}$$

Each user $a \in A$ participates in a set of interactions denoted by $D_a \subseteq D$, where D is the set of all interactions. Each interaction $d \in D_a$ contains a set of items W_d. Then each w is generated by a topic $z \in Z$, and z is in turn generated by

the topic distribution θ_a of user a.

$$z \sim Multinomial(\theta_a)$$
$$w|z \sim Multinomial(\phi_z)$$

In this static formulation, the problem is to find the posterior distribution $P(\phi_z|D, \beta), \forall z \in Z$ and $P(\theta_a, |D, \alpha), \forall a \in A$ given the set of interactions D.

Decay Topic Model

The above static model assumes that a user's topic distribution remains the same over time. However, in an evolving content network, a user may adopt content items of different topics over time. We extend the above notations to model the notion of temporality. Let T denote an ordered set of discrete time steps with order relation $<$ such that $\forall t_1, t_2 \in T$, $t_1 < t_2$ implies that t_1 is earlier than t_2. $\forall a \in A$, each user a has a topic distribution $\theta_{a,t}, \forall t \in T$, where $\theta_{a,t}$ is modeled as a Dirichlet Distribution.

$$\theta_{a,t} \sim Dirichlet(\{\alpha_{a,t,z}\}_{z \in Z})$$

Unlike the static topic model, each time step t has a Dirichlet distribution for the topic of user a parameterized by a set of parameters specific to the respective user and time. Since our focus here is on the evolution of users' topic distribution over time, to isolate its effects, we keep topic item distribution ϕ_z the same over time.

Each user $a \in A$ participates in a set of interactions in time step t as denoted by $D_{a,t} \subseteq D_t$, where D_t represents the set of interactions in time t. The interaction $d \in D_{a,t}$ contains a set of items W_d. Then each $w \in W_d$, w is generated by a topic $z \in Z$ and z is in turn generated by the topic distribution of user a at time t.

$$z \sim Multinomial(\theta_{a,t})$$

Hence, what is of interest to us now is the posterior distribution in each time step t, $P(\theta_{a,t}|D_t, \alpha), \forall a \in A, \forall t \in T$.

Generative Process: To arrive at this posterior distribution, we propose the *Decay Topic Model*, which we illustrate using the following generative process.

1. At time t, each user a samples their prior topic distribution $\theta_{a,t}$ from Dirichlet distribution with parameters $\{\alpha_{a,t,z}\}_{z \in Z}$.

2. User a samples the topic distribution $\phi_z, \forall z \in Z$ from Dirichlet distribution with symmetric parameters β.

3. For each interaction $d \in D_{a,t}$, there are a set of content items W_d. In turn, for each of the $|W_d|$ items:

   (a) User a generates a topic z_w from $\theta_{a,t}$ for the item w.

   (b) User a generates an item w from the topic item distribution ϕ_z.

4. Update the parameters of $\phi_z, \forall z \in Z$.

5. Update the parameters of $\theta_{a,t}$ to obtain the posterior topic distribution of a at time t. The posterior distribution also follows a Dirichlet distribution with parameters $\{\alpha_{a,t,z} + n_{a,t,z}\}, \forall z \in Z$, where $n_{a,t,z}$ denotes the number of items that user a adopted in time t that belongs to topic z.

6. For every $a \in A$, let the prior topic distribution of $t+1$ be the posterior distribution of t with the parameters multiplied by a decay factor, δ, such that $0 \leq \delta \leq 1$. i.e., $\alpha_{a,t+1,z} = \delta \times (\alpha_{a,t,z} + n_{a,t,z}), \forall z \in Z$, then the prior distribution $\theta_{a,t+1} = Dirichlet(\{\alpha_{a,t+1,z}\}_{z \in Z})$.

7. Repeat steps 1 to 6 for all the time steps.

Refer to Algorithm 1 for the outline of the inference procedure.

Decay Factor: The decay factor δ in step 6 helps to moderate the rate of change in topic preferences of users by balancing the contributions of the past

Algorithm 1 DTM Inference

{The first part is LDA inference taking into account of the interactions}
Input: Adoption data for each user a at each time step t
Output: Estimated parameters
{Initialization}
for $a \in A$ **do**
  {$D_{a,t}$ represents the set of interactions a has at time t}
  for $d \in D_{a,t}$ **do**
    {W_d represents the set of items adopted due to interaction d}
    for $w \in W_d$ **do**
      $k \leftarrow uniformRandom(1, |Z|)$
      {$n_{a,k}$ denote the number of times user a generated topic k.}
      {$m_{k,w}$ denote the number of times topic k generated item w.}
      $n_{a,k} \leftarrow n_{a,k} + 1$, $m_{k,w} \leftarrow m_{k,w} + 1, z_w \leftarrow k$
    end for
  end for
end for
{LDA Gibbs Sampling}
while iterate **do**
  for $a \in A$ **do**
    for $d \in D_{a,t}$ **do**
      for $w \in W_d$ **do**
        $k \leftarrow z_w, n_{a,k} \leftarrow n_{a,k} - 1$, $m_{k,w} \leftarrow m_{k,w} - 1$
        $k \leftarrow sample(n_{a,k} + \alpha, m_{k,w} + \beta)$
        $n_{a,k} \leftarrow n_{a,k} + 1$, $m_{k,w} \leftarrow m_{k,w} + 1, z_w \leftarrow k$
      end for
    end for
  end for
end while
{The second part obtains the topic distributions at different time step by decaying and conditioning on the learned latent variables in previous time step.}

for $a \in A$ **do**
  {T_a represents the set of a's active time steps}
  for $t \in T_a$ **do**
    if t is first time step **then**
      $n_{a,t,k} = \alpha$, for $k = 1$ to $|Z|$
    else
      $n_{a,t,k} = \delta \cdot n_{a,t-1,k}$, for $k = 1$ to $|Z|$
    end if
    for $d \in D_{a,t}$ **do**
      for $w \in W_d$ **do**
        $k \leftarrow z_w, n_{a,t,k} = n_{a,t,k} + 1$
      end for
    end for
  end for
end for

time steps versus the current time step. $\delta = 1$ implies no decay. $\delta = 0$ implies that we expect the authors to change their topic distribution at every time

step. In other words, by setting $0 \leq \delta \leq 1$, we want to adjust the importance of content adopted earlier compared with the recent content for determining the topic distribution of a. For instance, $\delta = 0.5$ means the preferences of a accumulated over time drops by half at every time step, i.e., the half life is one time step. The right setting of δ may differ in different scenarios. In the experiments, we conduct parameter sensitivity test to help determine the best δ setting. In the case where a user has no interaction at time t, her topic distribution will still remain the same as at previous time step $t-1$.

In this work, δ applies to the whole network. While it may be argued that δ may vary from user to user, and from time step to time step, in practice that would generate too many variables, which we may not be able to learn effectively.

4.2.2 Temporal Social Correlation Measure

Having modeled a user's changing topic distribution over time, we now investigate how to model a user's evolving social correlation with other users. This evolving temporal social correlation has been shown in the example as shown in Figure 4.3. In our formulation, the key idea is that, for user a to correlate heavily with another user c at time t, the following criteria have to be met:

- **Interactions.** User a participates in one or more interactions with c at time t. We assume that when an interaction between two users is observed at time t, the actual interaction would have taken place before t. This is reasonable given that our model works on time steps that combine interactions from several time points.

- **Content change.** User a's topic distribution grows to resemble c's topic distribution in the previous time step, i.e., between time steps $t-1$ and t, a's topic is becoming more similar to c's.

Based on the above principles, we propose the *Temporal Social Correlation Measure* in the form of a vector $\mathbf{I}_{a,t}$, which is computed as follows.

Given :

1. The set of interactions $D_{a,t}$ that user a participates at time t.
2. Topics associated with the content items, i.e., $\{z_w \mid w \in \bigcup_{d \in D_{a,t}} W_d\}$.
3. Topic distribution $\theta_{c,t-1}$ of every user $c \in \bigcup_{d \in D_{a,t}} A_d$, who has participated in at least one interaction with a in the previous time step $t-1$.

Find : Correlation vector $\mathbf{I}_{a,t}$, where each element $I_{a,t}(c)$ is the directed correlation of a to user $c \in \bigcup_{d \in D_{a,t}} A_d$.

Algorithm :

1. Initialize the array $\mathbf{I}_{a,t}$ with zero elements.
2. For each interaction $d \in D_{a,t}$, content item $w \in W_d$, and user $c \in A_d$,
   (a) We determine the generation of topic z_w by a user c as follows:

$$P(z_w|c, \theta_{c,t-1}) \propto \theta_{c,t-1,z_w}$$

   (b) Then update array $\mathbf{I}_{a,t}$ as follows,

$$\begin{aligned} I_{a,t}(c) &= I_{a,t}(c) + P(z_w|c, \theta_{c,t-1}) \\ &= I_{a,t}(c) + \frac{\theta_{c,t-1,z_w}}{\sum_{c \in A_d} \theta_{c,t-1,z_w}} \end{aligned}$$

3. Normalize the array $\mathbf{I}_{a,t}$ to sum to one for easy interpretation.

Step 2(b) calculates the contribution of each item w and its corresponding topic z_w to user a's correlation to user c. The higher the probability of c generating this topic, the higher is the value of $I_{a,t}(c)$. We assume that the

generation of topic z_w comes from a linear combination of a's friends and a herself. The correlation of a to c should be proportional to how much c is likely to generate the topic z_w. The correlation also accounts for the frequency of interaction, i.e. the more interactions a has with c, the higher is the value of $I_{a,t}(c)$.

We run this computation chronologically for every time step $t = 1$ to T to obtain the temporal social correlation values $I_{a,t}(c)$ for each a and c across different time steps $t \in T$.

We explore how the temporal social correlation measure is affected by the topic modeling of content network. At each time step t, we want to compare the changes of a's topic distribution $\theta_{a,t}$ and the changes of c's topic distribution $\theta_{c,t-1}$ for every c in $\bigcup_{d \in D_{a,t}} A_d$. Note that this set of users that a interacts with also contains a herself. Without any decay factor in the topic modeling, the accumulative effect over time will favor larger self-dependency values for a. The decay factor acts to reduce the importance of topics in previous time steps, allowing new interactions to change the topic distribution of a in t significantly enough, so as to better detect a's temporal social correlation with others.

4.3 Experiments

While user interaction networks model many kinds of interactions, there are only a limited number of datasets available for research, which track those interactions over a significant period of time. We work with two such datasets derived from DBLP and ACM Digital Library (ACM DL). We model co-authorship as a user interaction network, where a publication d is an interaction between one or more authors a_i(s) in the year t. The content items w associated with d are words in the titles/abstracts. In this setting, we say author a has temporal social correlation with author c, if a and c co-author a paper (interact) on topics that a is unlikely, but c is likely, to publish. We

assign temporal social correlation to co-authors of a based on the likelihood of the co-authors generating the topics in the papers that a publishes.

After describing the datasets, we will first evaluate the *Decay Topic Model* by comparing two settings (decay vs. non-decay) on the task of predicting an author's observed topic distribution in the next time step. We then evaluate: *Temporal Social Correlation Measure*, by conducting two prediction tasks. The first task is similar to the above but with a different approach. Instead of using an author's own topic distribution, we use her co-authors', weighted by the author's temporal social correlation with each co-author. The second task predicts an author's ranking of her co-authors by topic similarity at the next time step using tempoal social correlation at the current time step.

4.3.1 Datasets

For experiments, we use a subset of publications from DBLP and ACM DL. To ensure a wide coverage of fields in Computer Science, we use papers published in the reputable Journal of ACM (JACM) as a seed set. We grow this seed set by including other non-JACM publications by authors who has at least one JACM publication. We extend this further to also include the co-authors of JACM authors, and their publications as well.

Table 4.1: Dataset Sizes

	#authors	*#papers*	*#unique non-stop words*	*period*
DBLP	268,299	546,500	83,440	1936–2011
ACM	157,693	188,086	217,667	1952–2011

The sizes of our datasets are given in Table 4.1. DBLP has almost three times as many publications as ACM DL. One reason is the longer history of publications maintained by DBLP (since 1936). Another is the larger scope, since ACM DL focuses mainly on ACM-related publications. However, ACM DL has many more unique words than DBLP, because ACM DL has both titles and abstracts, whereas DBLP only has titles. In both cases, the datasets are

significantly large, with hundreds of thousands of nodes, with more than 10 million author-word links for DBLP and 46 million author-word links for ACM DL.

Table 4.2: Top Words for Sample Topics

Web Systems and Algorithms	*Computational Biology*	*Database Systems and Theory*
DBLP		
web	protein	data
information	gene	database
semantic	analysis	query
based	data	xml
retrieval	database	processing
ACM		
web	data	data
information	gene	query
search	protein	database
content	biological	xml
user	expression	processing

To show that topic modeling on these datasets would discover the latent topics effectively, we produce three sample topics, and the top words for each topic of DBLP and ACM in Table 4.2. Notably, the top words (e.g., web, information, retrieval) capture well the essence of the topics (e.g., Web systems and algorithms). Moreover, both DBLP and ACM DL discover similar topics with similar top words, even when DBLP has only titles and ACM DL has both titles and abstracts. During experiments, we observe that both datasets result in similar observations. From here onwards, we will use the larger dataset DBLP as the main dataset to discuss our results.

4.3.2 Evaluating Decay Topic Model

The topic modeling step seeks to arrive at $\theta_{a,t}$, the topic distribution of each author a at time t. We hypothesize that the decay factor allows it to better adapt to the author's changing preferences over time. Without decay, the accumulative effect tends to overweigh the older topics more heavily. To test

this hypothesis, we compare the topic distributions $\delta < 1$ ($\theta_{a,t}^{decay}$) and $\delta = 1$ ($\theta_{a,t}^{non-decay}$) to the observed topic distribution at the next time step ($\theta_{a,t+1}^{obs}$). For $\theta_{a,t}^{decay}$, we vary δ from 0.2 to 0.8 to determine the optimal setting of δ.

Both $\theta_{a,t}^{decay}$ and $\theta_{a,t}^{non-decay}$ incorporate information from the first time step to the current time step t. We compare them to $\theta_{a,t+1}^{obs}$, which is derived independently using only the set of documents published by a at time $t+1$. If $\theta_{a,t}^{decay}$ is more similar to $\theta_{a,t+1}^{obs}$ than $\theta_{a,t}^{non-decay}$, it shows that the decay approach is better adapted to the preferences in $t+1$.

To measure similarity between two probability distributions p and q, we use the following function:

$$Sim(p, q) = 1 - D_{JS}(p, q),$$

where D_{JS} is the Jensen-Shannon Divergence [68]. Sim ranges from 0 (different) to 1 (identical).

We use this Sim function to measure the similarity between $\theta_{a,t}^{decay}$ and $\theta_{a,t}^{non-decay}$ respectively to $\theta_{a,t+1}^{obs}$. To compare the decay vs. non-decay setting directly, we then take the ratio of the two similarity values as follows:

$$\text{Sim Ratio } \Phi(a, t) = \frac{Sim(\theta_{a,t}^{delta}, \theta_{a,t+1}^{obs})}{Sim(\theta_{a,t}^{non-decay}, \theta_{a,t+1}^{obs})}$$

$\Phi(a, t)$ ratio > 1 indicates that having the decay is better than no decay. Figure 4.4 shows the mean of values given by the Sim Ratio $\Phi(a, t)$ with respect to the different δ values. Given that the values lie above 1, it indicates that that having some decay is better than no decay at all. From the various choices of δ, we can see that the optimal value of δ lies between 0.4 to 0.5. For the rest of our experiments we therefore use $\delta = 0.5$.

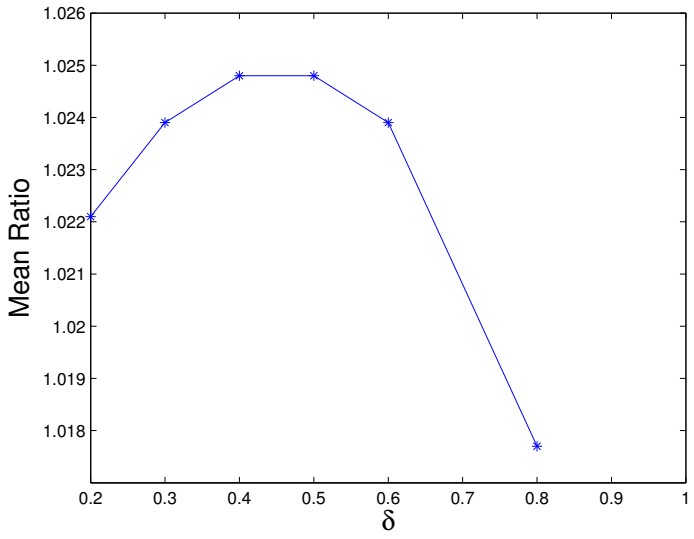


Figure 4.4: DBLP: Mean of Sim Ratio $\Phi(a,t)$

4.3.3 Prediction of Author's Topic Distribution

We now show that our correlation values at t can also be used for the prediction of author a's topic distribution in $t+1$. In this case, the predicted topic distribution for a at time $t+1$ will be a linear combination of the topic distributions at time t of her co-authors $c \in \bigcup_{d \in D_{a,t}} A_d$, weighted by the temporal social correlation values $I_{a,t}(c)$.

Hence, if one set of correlation values arrive at a better estimation of the author's topic distribution than another set of correlation values, it implies that the former more accurately estimate the temporal social correlation weight of each co-author.

Due to the way in which we extract the subset of data from DBLP and ACM DL, we can only evaluate for the authors who have at least one JACM paper. For these authors, we have the complete co-authors information, while for the rest of the other authors, we have only partial information.

Decay vs. Non-decay: To evaluate our topic prediction, we use the Sim Ratio $\Phi(a,t)$ as defined earlier to compare against the observed topic distribution at $t+1$ (based on only the documents published at time $t+1$) as ground truth. The first comparison is again for decay vs. non-decay, but this

time the prediction is based not on the author's own topic distribution, but rather on her co-authors'. We derive two predicted topic distributions at $t+1$, $\theta^{dep-d}_{a,t+1}$ and $\theta^{dep-nd}_{a,t+1}$. $\theta^{dep-d}_{a,t+1}$ is computed using the correlation value $I^{decay}_{a,t}(c)$, for each $c \in \bigcup_{d \in D_{a,t}} A_d$ by the decay topic distribution. $\theta^{dep-nd}_{a,t+1}$ is computed using the correlation values $I^{non-decay}_{a,t}(c), c \in \bigcup_{d \in D_{a,t}} A_d$ by the non-decay topic distribution.

User a's predicted preference for topic z at time $t+1$ is computed as follows.

$$\theta^{dep-d}_{a,t+1,z} = \sum_{c \in \bigcup_{d \in D_{a,t}} A_d} I^{decay}_{a,t}(c) * \theta^{decay}_{c,t,z}$$

$$\theta^{dep-nd}_{a,t+1,z} = \sum_{c \in \bigcup_{d \in D_{a,t}} A_d} I^{non-decay}_{a,t}(c) * \theta^{non-decay}_{c,t,z}$$

We then compute the Sim Ratio $\Phi_1(a,t)$ as follows.

$$\text{Sim Ratio } \Phi_1(a,t) = \frac{Sim(\theta^{dep-d}_{a,t+1}, \theta^{obs}_{a,t+1})}{Sim(\theta^{dep-nd}_{a,t+1}, \theta^{obs}_{a,t+1})}$$

$\Phi_1(a,t)$ ratio > 1 would indicate that the correlation values computed by decay topic distribution give a better prediction than the correlation values computed by the non-decay topic distribution. Figure 4.5(a) shows a histogram of $\Phi_1(a,t)$ values. The x-axis of the histogram are bins with boundaries given by the value of $\Phi_1(a,t)$. The y-axis of the histogram indicate the frequency of author a and time point t pairs falling into the respective bins. For Figure 4.5(a), 68% of the (a,t) pairs have $\Phi_1(a,t) > 1$, 1% have $\Phi_1(a,t) = 1$ and 31% have $\Phi_1(a,t) < 1$. This suggests that incorporating the decay factor results in an improvement for the large majority of (a,t) pairs.

Correlation vs. Co-authorship Count: As another baseline, we use a naive way of computing temporal social correlation weight $I^{base}_{a,t}(c), c \in \bigcup_{d \in D_{a,t}} A_d$, which considers a's correlation with c at t as the count of papers

co-authored by a and c, normalized by the total count of a's papers, at time t. Using such temporal social correlation weights, we compute the predicted topic distribution $\theta^{base}_{a,t+1}$, and compare this to the correlation-based prediction $\theta^{dep-d}_{a,t+1}$.

$$\theta^{base}_{a,t+1,z} = \sum_{c \in \bigcup_{d \in D_{a,t}} A_d} I^{base}_{a,t}(c) * \theta^{decay}_{c,t,z}$$

For this comparison, we compute the Sim Ratio Φ_2 as follows.

$$\text{Sim Ratio } \Phi_2(a,t) = \frac{Sim(\theta^{dep-d}_{a,t+1}, \theta^{obs}_{a,t+1})}{Sim(\theta^{base}_{a,t+1}, \theta^{obs}_{a,t+1})}$$

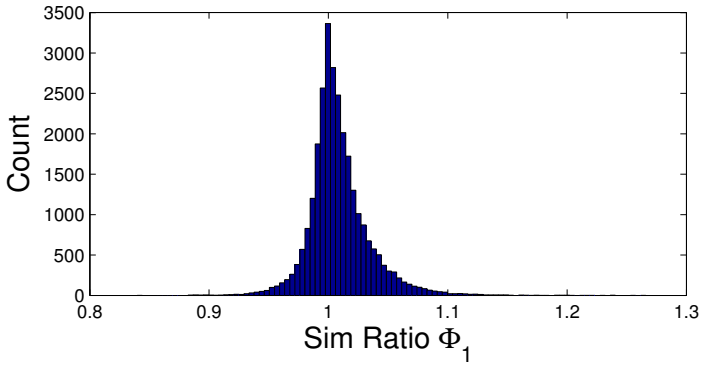
$\Phi_2(a,t) > 1$ indicates that the correlation method outperforms the baseline. Figure 4.5(b) shows a histogram of $\Phi_2(a,t)$ values. In the figure, 62% have $\Phi_2(a,t) > 1$, 1% have $\Phi_2(a,t) = 1$ and 37% have $\Phi_2(a,t) < 1$. This implies that in most cases, the correlation method tends to arrive at a better prediction than the co-authorship baseline.

Result Analysis: We now seek to understand better the profiles of users for which our method works especially well. As mentioned previously, Sim Ratio $\Phi(a,t) > 1$ indicates that our method performs better than the baseline at predicting an author's topic distribution. Most authors are active for more than one year, and each year gives a different Sim Ratio. Hence, the proportion of years in which $\Phi(a,t) > 1$ for a given author indicates the degree to which the user has benefited consistently from our proposed method.

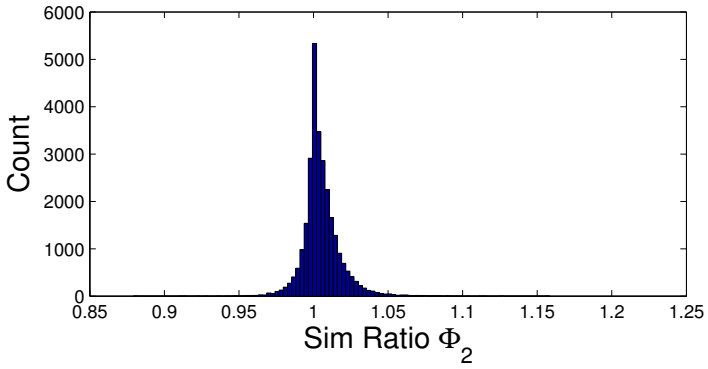
To measure this, we introduce the following metric:

$$\Psi(a) = \frac{\text{number of years where } \Phi > 1 \text{ for } a}{\text{number of years } a \text{ publishes}}$$

Figure 4.6(a) shows the histogram of $\Psi(a)$ values for various users, for a comparison against the non-decay baseline (i.e., $\Phi_1(a,t) > 1$). Figure 4.6(b) shows the corresponding histogram of $\Psi(a)$, for a comparison against the co-authorship baseline (i.e., $\Phi_2(a,t) > 1$). The red line in both figures indicate



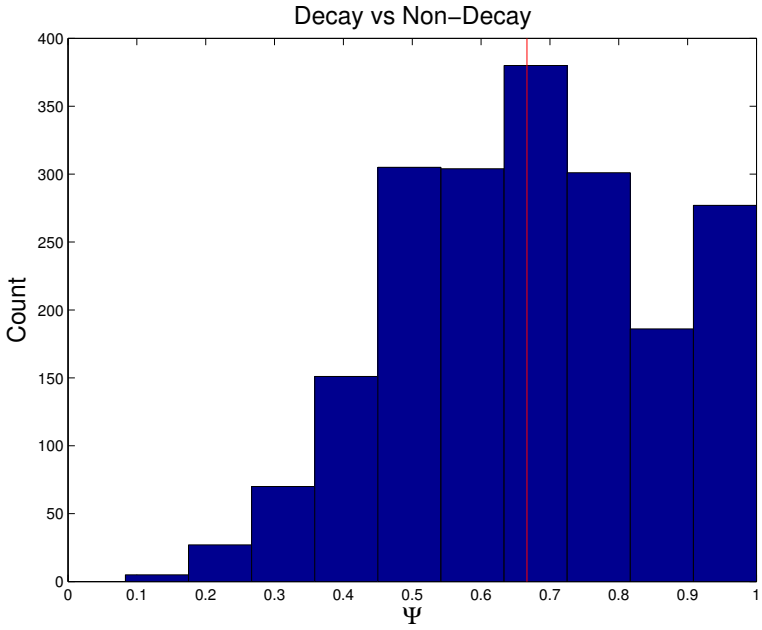
(a) DBLP: Histogram for Sim Ratio $\Phi_1(a,t)$



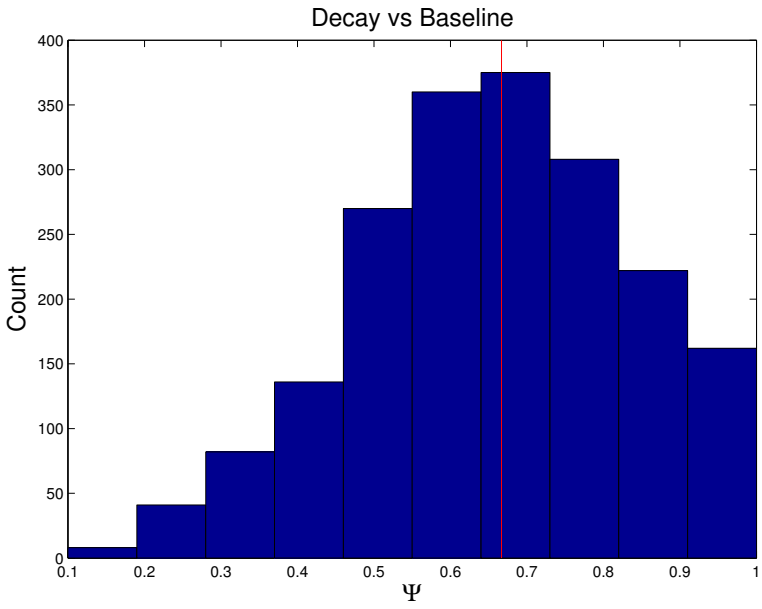
(b) DBLP: Histogram for Sim Ratio $\Phi_2(a,t)$

Figure 4.5: DBLP: Histogram for Sim Ratio $\Phi_1(a,t)$ & $\Phi_2(a,t)$

the median value of $\Psi(a)$ among the authors. In both cases, the median lies close to 0.7, which implies that a majority of users benefit from our proposed method at least two thirds of the time. In order for us to understand why we are able to predict the topic distribution of some authors and not the others, we examine the $\Psi(a)$ of each author with respect to some factors. Figures 4.7, 4.8 and 4.9 show the boxplots of $\Psi(a)$ with respect to their number of active years, the total number of papers published and the number of co-authors they worked with over the entire duration of their careers. The bins in the boxplots are determined by having equal number of data points and the labels on the x-axis represent the mean value of the data points in each bin. The figures collectively tell the story of better performance for authors with higher number of active years, papers, and co-authors. This suggests that we tend to do better when there is more information for a given author. The consistency



(a) DBLP: Histogram for $\Psi(a)$ against non-decay baseline



(b) DBLP: Histogram for $\Psi(a)$ against co-authorship baseline

Figure 4.6: DBLP: Histogram for $\Psi(a)$

and the degree to which an author interacts with others allow better inference of not just their topic distributions, but also their temporal social correlation values.

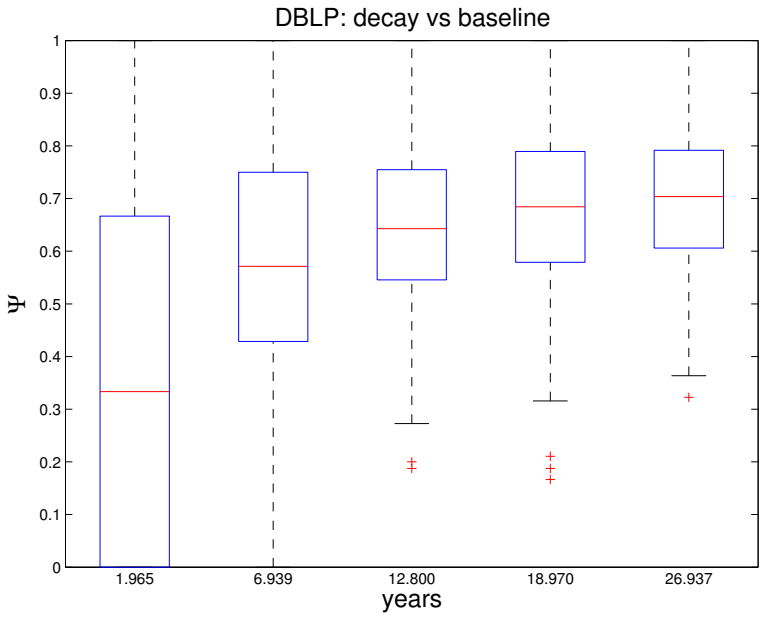


Figure 4.7: DBLP: $\Psi(a)$ vs Number of Active Years

4.3.4 Prediction of Co-Author's Topic Similarity Ranking

In this section, we perform co-author's topic similarity ranking prediction at time $t+1$ using temporal social correlation at time t. At time t, an author a has temporal social correlation value of $I_{a,t}(c)$ with a co-author c. Assuming that a usually does not change the temporal social correlation with her co-authors drastically over two time steps, we expect the ranking of her co-authors by temporal social correlation at time t would be a good predictor for the ranking at time $t+1$. Since a does not necessarily have identical sets of co-authors in t and $t+1$, the ranking prediction will only involve the co-authors appearing in both t and $t+1$. Similar to the previous experiment, we only evaluate for authors who have at least one JACM paper.

In this task, we denote the *ground truth* ranking of co-authors by topic similarity as R_{sim}. We derive R_{sim} for an author a at time $t+1$ as follows. For each co-author c of a, we obtain the "observed" topic distribution $\theta^{obs}_{c,t+1}$ using only publications by c at time step $t+1$. We then compute the similarity between c's topic distribution $\theta^{obs}_{c,t+1}$ with author a's observed topic distribution

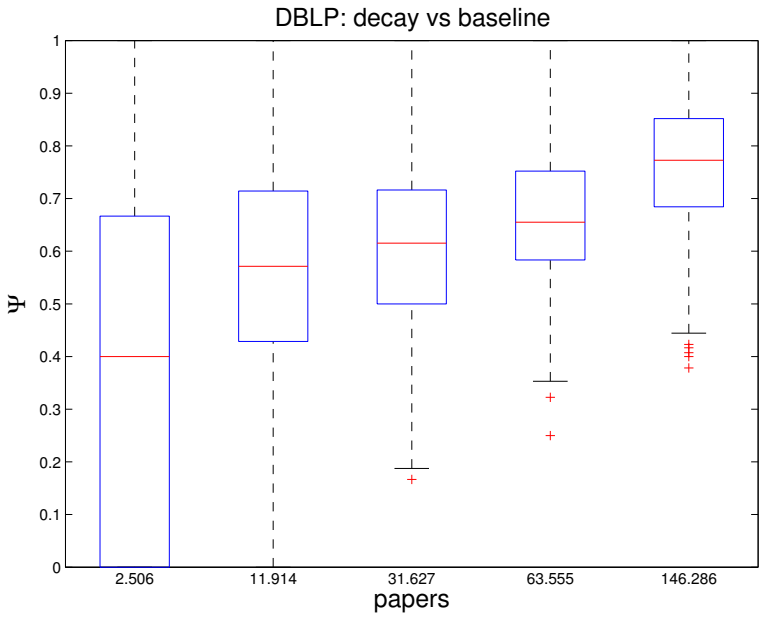


Figure 4.8: DBLP: $\Psi(a)$ vs Number of Published Papers

$\theta^{obs}_{a,t+1}$ using the *Sim* function as defined earlier in Section 4.3.2. Finally, we obtain the ranked list by sorting a's co-authors in descending order of the similarity values.

We compare the R_{sim} of a at time $t+1$ (ground truth) with the following two ranked lists:

1. *Temporal Social Correlation:* R_{dep} ranks co-authors in terms of $I_{a,t}(c)$.

2. *Co-authorship Baseline:* R_{base} ranks co-authors in terms of the number of co-authored papers at time t.

We derive the pair-wise rank correlations between R_{dep} (or R_{base}) and R_{sim} using Kendall Tau Rank Correlation Coefficient (tau coefficient) [53], which is a measure of correlation between two ranked lists where 1 represents full positive correlation, -1 represents full negative correlation and 0 represents no correlation. Hence, if $tau(R_{dep}, R_{sim})$ is higher than $tau(R_{base}, R_{sim})$, it implies that the proposed temporal social correlation measure has higher predictive value than the baseline co-authorship method. To perform this comparison, we first compute the R_{sim}, R_{dep}, and R_{base} for all authors. We then bin the authors into five equisized bins according to their $tau(R_{dep}, R_{base})$. The bin

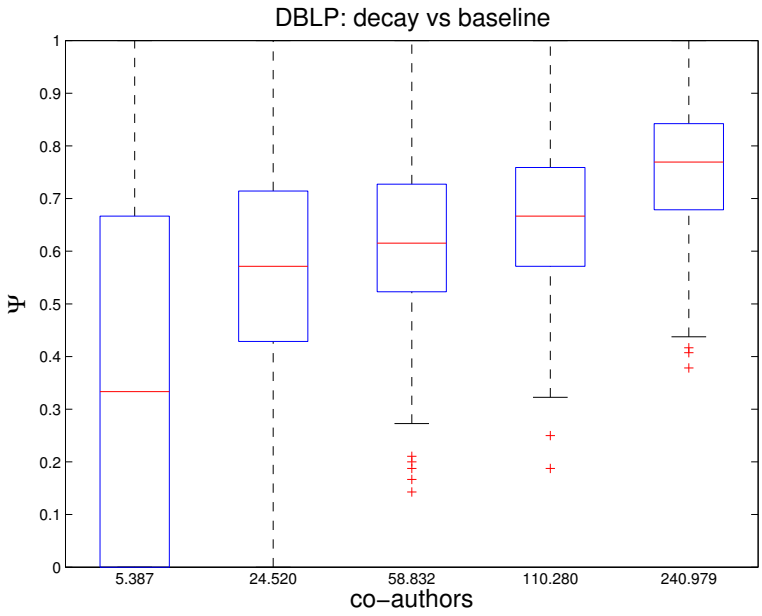
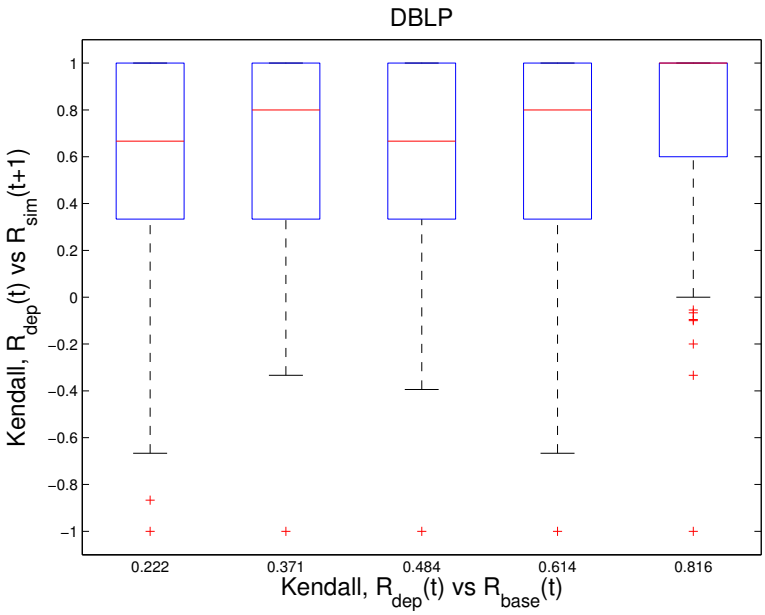


Figure 4.9: DBLP: $\Psi(a)$ vs Number of Co-Authors

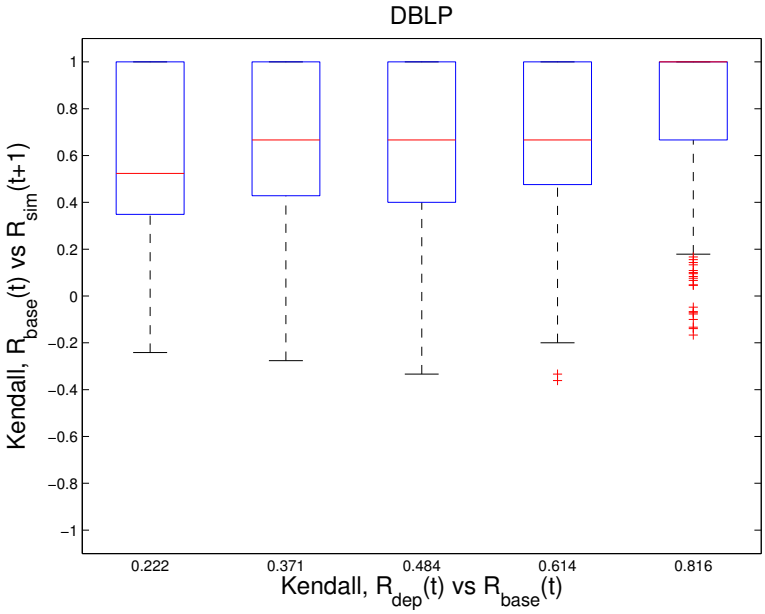
with the smallest values group authors for which R_{dep} and R_{base} are most different. The bin with the highest values group authors for which R_{dep} and R_{base} are most similar.

We then look at the distribution of $tau(R_{dep}, R_{sim})$ values in each bin. Figure 4.10(a) shows a boxplot representation of $tau(R_{dep}, R_{sim})$ distributions (y-axis) for each of the five $tau(R_{dep}, R_{base})$ bins (x-axis). The number shown in the x-axis is the mean within each bin. The red line in each box represents the median value, edges of the blue box represents the 25^{th} and 75^{th} percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually. Figure 4.10(b) shows the corresponding boxplot representation for the baseline $tau(R_{base}, R_{sim})$.

Comparing Figure 4.10(a) (proposed) and Figure 4.10(b) (baseline), we observe that for each bin, the boxplots in Figure 4.10(a) consistently show higher medians (higher similarity to the ground truth) than the boxplots in Figure 4.10(b). As the previous figures capture only the DBLP dataset, we repeat a similar experiments for the ACM dataset as well. The results for ACM are given in Figures 4.11(a) and 4.11(b), where similar observations can



(a) Evaluating Correlation-based Ranking



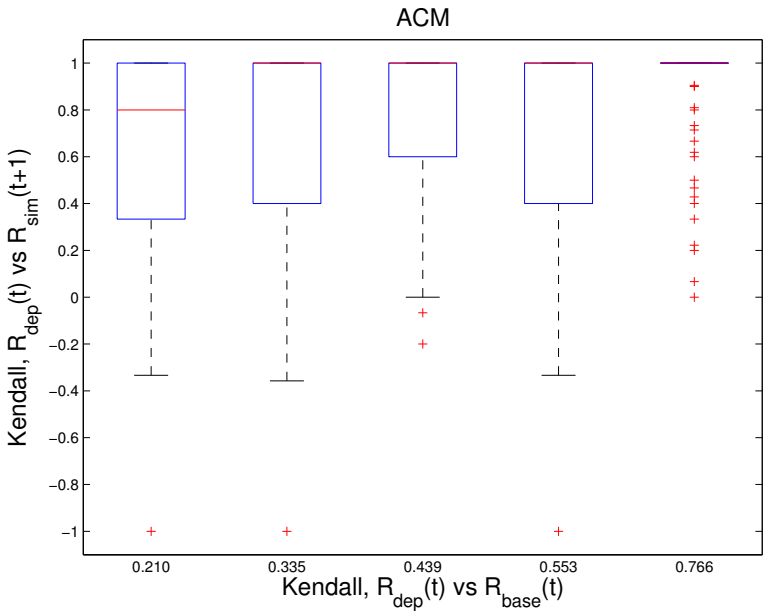
(b) Evaluating Baseline Co-authorship-based Ranking

Figure 4.10: DBLP: Evaluating Ranking Results

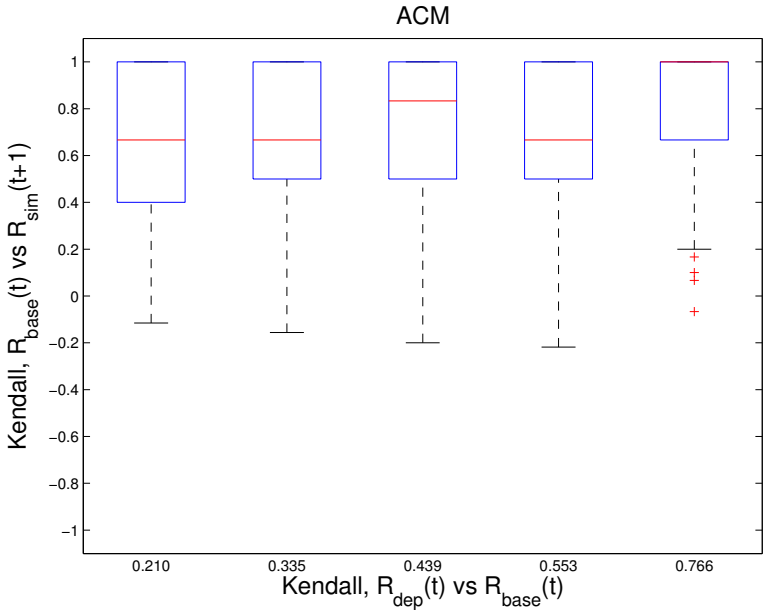
be made to support the higher prediction performance of R_{dep}, as compared to the baseline R_{base}.

4.3.5 Case Study

Using DBLP, we provide a case study to help illustrate the workings of our proposed temporal social correlation model. For this case study, we use the



(a) Evaluating Correlation-based Ranking



(b) Evaluating Baseline Co-authorship-based Ranking

Figure 4.11: ACM: Evaluating Ranking Results

profile of Associate Professor Duminda Wijesekera. Figure 4.12 shows the temporal social correlations of Duminda Wijesekera for the year 2001. The directed edges show Duminda Wijesekera's correlations with his co-authors who publish with him in the year 2001. Next to these co-authors are their respective topic distributions for year 2000. From the year 2000 to 2001, we observed that Duminda Wijesekera's topic in Security has increased from third

position to first position [112]. Based on the correlations, we observe that he correlated with Sushil Jajodia most as compared to other co-authors (excluding himself). Based on the co-authors topic distribution, Sushil Jajodia's topic in Security is the highest which explains why Duminda Wijesekera's correlation with Sushil Jajodia is the highest [20]. In 2002, Duminda Wijesekera continues to increase his topic in Security [108, 113]. This illustrates how temporal social correlation works based on the two components of interactions as well as content change.

4.4 Summary

In this chapter, we address the problem of modeling the evolution of user interaction networks, in order to determine the temporal social correlation weights among users at various time steps. We identify two primary factors to temporal social correlation, namely: interactions between users, and temporal correlation between the users' topic distributions. We propose a *Decay Topic Model* to model a user's evolution of content at the topic level, as well as a *Temporal Social Correlation Metric* to determine the degree to which a user is correlated with another user. Comprehensive experiments on real-life co-authorship datasets DBLP and ACM show that our proposed models perform well against the baseline (co-authorship count) in two predictive tasks: predicting an author's ranking of co-authors by temporal social correlation, as well as predicting the author's topic distribution in the next time step. This validates our hypothesis that we also need to take into account the changing topic preferences of users beyond just interactions (which the co-authorship baseline only models indirectly).

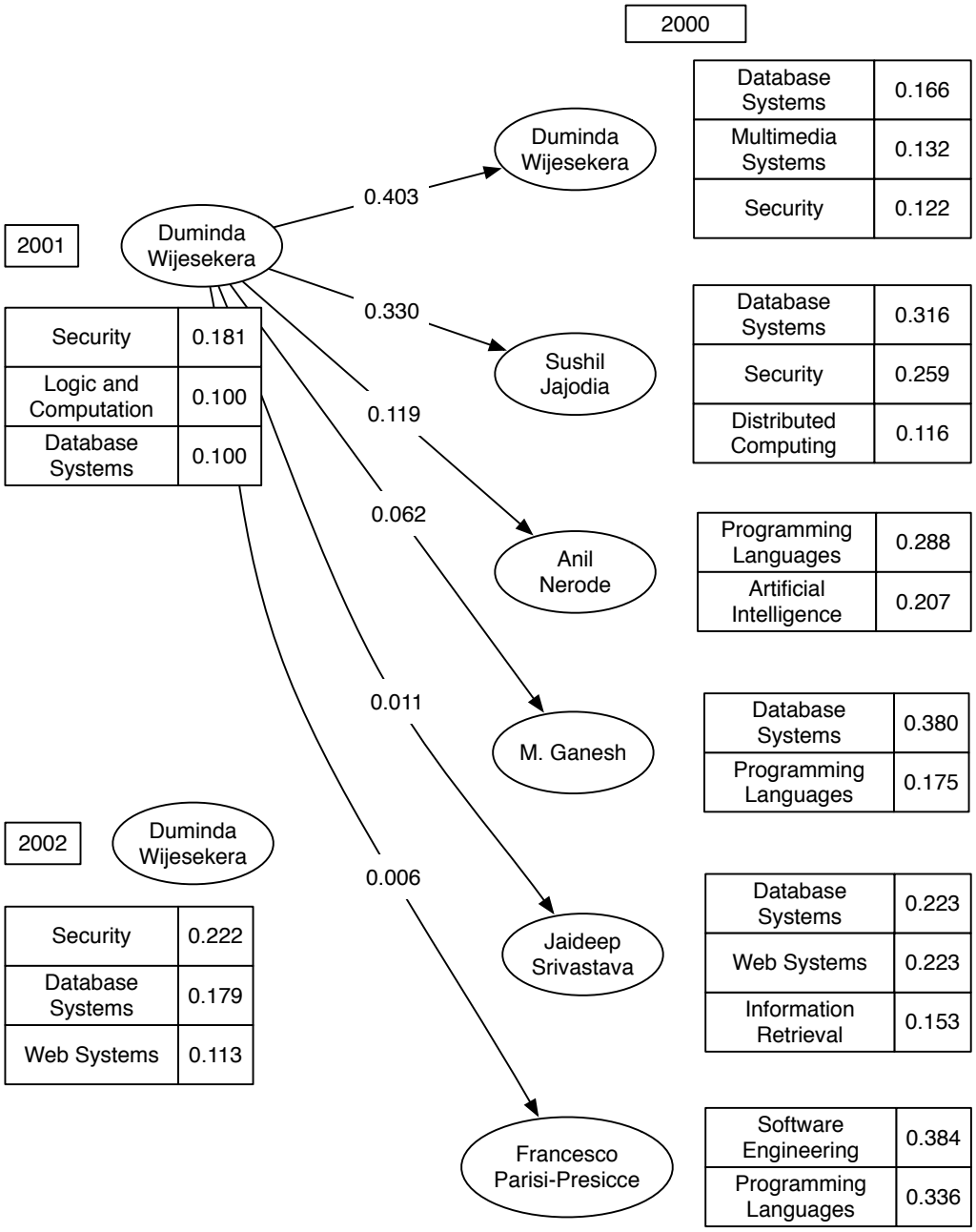


Figure 4.12: DBLP Case Study

This page was intentionally left blank.

Chapter 5

Dynamic Matrix Factorization for Modeling Temporal Adoptions

In this chapter, we adopted a Dynamic Matrix Factorization (DMF) technique to derive different temporal factorization models that can predict missing adoptions at different time steps in the users' adoption history. This DMF technique is an extension of the Non-negative Matrix Factorization (NMF) based on the well-known class of models called Linear Dynamical Systems (LDS). By evaluating our proposed models against NMF and TimeSVD++ on two real datasets extracted from ACM Digital Library and DBLP, we show empirically that DMF can predict adoptions more accurately than the NMF for several prediction tasks as well as outperforming TimeSVD++ in some of the prediction tasks. We further illustrate the ability of DMF to discover evolving research interests for a few author examples.

5.1 Adoption Modeling

Recommender systems have been widely used to suggest products, content and services to consumers. Recommender techniques have been largely related

to rating prediction and evaluated on Netflix and Movielens datasets. The common assumptions underlying rating prediction are that: (a) each item can be rated or adopted only once by a user; (b) ratings assigned to items are restricted to a pre-defined rating options, say 1 to 5; and (c) the user-rate-item data is static. Although these assumptions are reasonable in many application settings, there are also many other settings that violate these assumptions.

For example, there are many application scenarios where a user can adopt the same item more than once, i.e. a user may buy the same product in different purchases. These include food, stationery, drug, and other items. A user may visit the same restaurant, bookstore, or cinema multiple times. In the context of social media, a user may adopt the same URL, tag or keyword multiple times as the user shares messages with her friends. When the same item is adopted at different time steps, the user may adopt it with different quantities as the user's preference or demand on the item changes over time. Assumption (a) hence does not hold in these scenarios and we need to consider recommending the same item even if it has been previously adopted.

The above scenarios also violate assumption (b) as they do not necessarily involve users giving ratings to items. The user's propensity to adopt an item can be measured by adoption quantity, which can be any non-negative integer value instead of a fixed range of rating values. A user may choose not to adopt an item at all if he dislikes the item, or adopt an item with a large quantity if he likes it. Adoption count also does not imply likeness. By adopting one instance of item does not mean the user does not like the item. Conversely, by adopting multiple instances of an item does not mean the user likes the item.

The last assumption (c) is clearly not applicable to many recommender systems involving dynamic user adoption patterns. These recommender systems have to determine trends that affect user adoptions. Unfortunately, most existing recommendation algorithms only deal with static adoption data. When applied to dynamic adoption data, the data is usually first divided into time

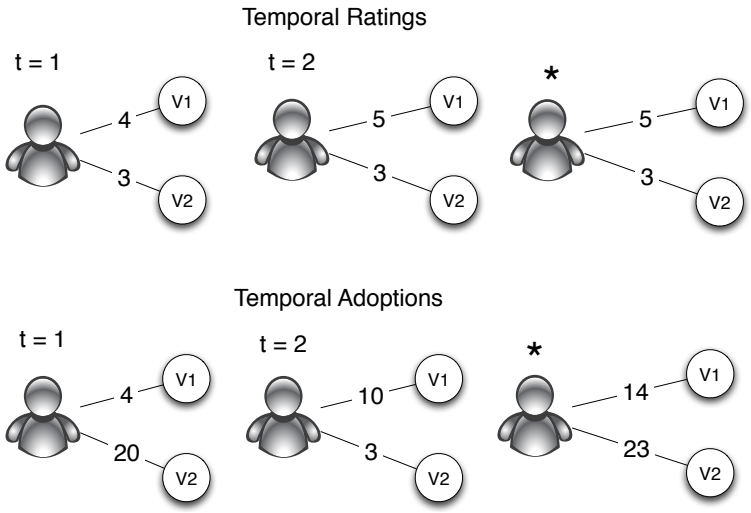


Figure 5.1: Temporal Rating vs Temporal Adoption

steps and the recommendation algorithm is applied to the adoption data in each time step independently of other time step. The result is that items recommended to a user in one time step may look entirely different from those recommended in the next time step, which is not ideal in many application settings.

Figure 5.1 shows an example between the differences of rating and adoption in two time steps $t = 1, 2$. In the case of temporal rating, the user can only rate an item once, any changes in the rating between the user and the item at a later time step is seen as an updated rating. When we collapse the data into its static equivalent (denoted by *), the value between user and item reflects the latest rating. However for temporal adoption, the edges in the collapsed static data (*) has weights that are aggregated through time.

In this chapter, we focus on addressing the problem of modeling users adopting items across different time steps to generate recommendations considering evolving user preferences. Unlike rating-based recommendation, we assume the same users can adopt items more than once with different quantity numbers.

The main idea of our approach is to model dynamic adoption data using a combination of *Non-negative Matrix Factorization* (NMF) and *Linear Dy-*

namical Systems (LDS). We represent the adoption data of each time step as a state defined by the preferences of users and the characteristics of items in low rank factors as well as transitions of low rank factors so as to smoothen the evolution of user preferences.

Suppose we model users adopting items as a bipartite graph where users and items represent the two types of vertices, the weights on the user-adopt-item edges represent the number of times the users adopt the items. For the time steps of adoption data, we can define a bipartite graph Y_t for each time step t. When a user n adopts w instances of an item m in time t, an edge is created between n and m and an edge weight w is assigned. In the adjacency matrix representation, this translates to $y_{m,n,t} = w$.

We now define the *dynamic adoption prediction problem* as follows. Given the item adoption data for a set of N users and M items in different time steps for $t = 1 \cdots T$, we want to find the low rank factors of user preference for every time step and to use the low rank factors to predict for the possibility of missing adoptions in each time step t.

A direct and simple way of solving dynamic adoption prediction problem is to perform NMF independently for each time step. Suppose we have M items and N users, using MF for K latent factors,

$$Y_t = C_t \cdot X_t$$

where $Y_t \in \mathbf{R}^{M \times N}$, $C_t \in \mathbf{R}^{M \times K}$ and $X_t \in \mathbf{R}^{K \times N}$.

But there are drawbacks to such an approach. Given that solving for NMF is a non-convex optimization problem, this approach suffers from the identifiability problem where multiple solutions exist. This makes the interpretation of resultant predictions difficult, as the lower rank factors are not related across different time steps. That is, the user preference factors derived for one time step may be completely unrelated with those for another time step.

Linear Dynamical Systems (LDS) offer an elegant way of expressing the

relationship between latent factors at different time steps. For each user n, LDS derives for each time step t a dynamics matrix $A_{n,t}$ that represents the mapping of latent factors from time step $t-1$ to t. When LDS is applied to a set of users, we obtain Dynamic Matrix Factorization (DMF). Different matrix factorization techniques can be utilized in DMF and this chapter introduces DMF based on NMF, a MF technique very often used for rating prediction. To the best of our knowledge, using LDS and NMF for DMF to model dynamic adoption data is novel and has not been attempted before. In a previous work by Sun et al., a Dynamic Matrix Factorization approach based on LDS has been developed for rating prediction [99], but they do not include the use of NMF. The use of NMF is extremely important for obtaining insights into the "topics" that users follow. Without the non-negativity constraints, the latent factors obtained for users become uninterpretable. Previous works on rating prediction [74, 55, 56, 57, 115] which employ Probabilistic Matrix Factorization (PMF) [89, 90] are not able to show interpretable topics because of the unconstrained sign of their latent factors. Our approach of enforcing a non-negativity constraint in DMF has never been applied and evaluated in item adoption prediction.

We briefly argue that non-negativity is necessary for ranking items in each latent factor to obtain interpretable topics. For a given element $y_{m,n}$ of the item-user matrix Y, the MF approach is to approximate $y_{m,n}$ using the item latent factors c_m and user latent factors x_n.

$$y_{m,n} = \sum_{k=1}^{K} c_{m,k} \cdot x_{k,n}$$

In topic models based on NMF [116, 69], the important items for each latent factor is obtained by ranking the items' value in the respective latent factor. So if $c_{i,k} > c_{j,k}$, it implies that item i is more representative than item j for the latent factor k in NMF. But this is not true if the latent factors contain

negative values. A negative $c_{m,k}$ can also be important for contributing to the value $y_{m,n}$ if the corresponding $x_{k,n}$ is also negative. Since the negative latent factors prevent one from interpreting their semantics, we propose to use NMF with LDS to obtain DMF with non-negative values.

These are the major highlights of this chapter:

- This chapter makes a clear distinction between item adoption recommendation and rating recommendation. We point out that as user interests evolve, we need to model these changes and adapt the prediction of adoption data temporally.

- We propose three evaluation tasks for comparing the performance of our proposed models against other baselines in the temporal item adoption problem.

- We conduct a series of experiments to show that our proposed models outperform NMF in the different prediction tasks and TimeSVD++ for some prediction tasks involving dynamic adoptions. A few author case examples illustrating changes of research interests learnt in DMF have also been given to highlight the knowledge discovered by using DMF.

We describe in detail our models in Section 5.2. Then Section 5.3 evaluates our models through empirical experiments. We finally end the chapter in 5.4.

5.2 Dynamic Matrix Factorization

Given the relationship between static Matrix Factorization (**MF**) and Dynamic Matrix Factorization (**DMF**), we show how to use the parameters obtained from the learning of MF for learning DMF.

5.2.1 Problem Definition

Dynamic adoption prediction can be formally defined as a MF problem for an *adoption matrix* $Y \in \mathbf{R}^{M \times N \times T}$, where M denotes the number of items, N denotes the number of users and T denotes the number of time steps. Each element $y_{m,n,t}$ of Y denotes the adoption count (≥ 0) for item m by user n in time step t.

Not all temporal adoptions in Y are observed. We denote $y_{m,n,t}$ as the temporal adoption observed for user n, item m and time step t. When $y_{m,n,t} = 0$, it means that the temporal adoption is *missing* or we do not observe the item n adopted by user m in the corresponding time step. Y is sparse as each user adopts usually only very few items.

The adoption matrix Y can be collapsed into a $M \times N$ *total adoption matrix* Y^* by aggregating the temporal adoptions of each user-item pair across all time steps. That is, each element of Y^* is obtained by $y^*_{m,n} = \sum_t y_{m,n,t}$.

Depending on what we want to predict for the adoption matrix Y, we can formulate three prediction tasks:

- Task 1, *Prediction of missing temporal adoptions*: The task of predicting missing adoptions at some time step t for some user n and item m and we represent the predicted adoptions by $\hat{y}_{m,n,t}$.

- Task 2, *Prediction of all total adoptions given missing temporal adoptions*: The task of predicting all total adoptions $y^*_{m,n}$ given some missing temporal adoptions, i.e., $y_{m,n,t} = 0$, **at some time step** t for some n and m for a set of (m, n, t) triplets. We represent the predicted total adoptions of user n and item m as $\hat{y}^*_{m,n}$

- Task 3, *Prediction of missing total adoptions*: The task of predicting missing total adoptions $\hat{y}^*_{m,n}$ for user n and item m with $y_{m,n,t} = 0$ **for all** $t \in T$.

We can solve the above prediction tasks in a naive approach using non-negative matrix factorization (NMF) in the next section before extending it to DMF.

5.2.2 Non-Negative Matrix Factorization

Given a static adoption matrix $Y^* \in \mathbf{R}^{M \times N}$, matrix factorization returns two lower ranked matrices item-factor matrix $C \in \mathbf{R}^{M \times K}$ and user-factor matrix $X^* \in \mathbf{R}^{K \times N}$, where K represents the number of factors. The *item-factor matrix* C represents the mappings from items to a set of factors, while the *factor-user matrix* X^* represents the mappings from factors to users. In adoption prediction tasks, we would like to regard the factor values as their weights and hence require them to be non-negative.

Non-negative matrix factorization (NMF) meets the requirement for non-negativity of both the item-factor matrix C and factor-user matrix X^*. NMF finds the lower rank matrices C and X^* such that their product recovers missing values in Y^*. As NMF is a well-defined and well-understood technique, we only briefly show how to solve for C and X^* using stochastic gradient descent (SGD) with the log-barrier approach for non-negativity constraints.

The parameters of NMF can be obtained using the following derivatives executed using multiple iterations,

$$\frac{\partial \log p(y^*_{m,n})}{\partial c_{m,k}} = \gamma \left(y^*_{m,n} - \sum_{k=1}^{K} c_{m,k} x^*_{k,n} \right) x^*_{k,n} + \frac{\xi}{c_{m,k}}$$
$$\frac{\partial \log p(y^*_{m,n})}{\partial x^*_{k,n}} = \gamma \left(y^*_{m,n} - \sum_{k=1}^{K} c_{m,k} x^*_{k,n} \right) c_{m,k} + \frac{\xi}{x^*_{k,n}}$$

$$new\ c_{m,k} = old\ c_{m,k} + \eta \cdot \frac{\partial \log p(y^*_{m,n})}{\partial c_{m,k}}$$
$$new\ x^*_{k,n} = old\ x^*_{k,n} + \eta \cdot \frac{\partial \log p(y^*_{m,n})}{\partial x^*_{k,n}}$$

where γ represents the precision of error, ξ represents the strictness of the log barrier constraint and η represents the rate of learning for SGD. In our experiments, we use the parameter settings $\gamma = 1$, $\xi = 0.01$, and $\eta = 0.0001$.

5.2.3 Dynamic Matrix Factorization

DMF can be seen as an extension of NMF by adding the time dimension based on *Linear Dynamical Systems* (LDS). LDS is originally designed to relate an output signal $y_t \in \mathbf{R}^M$ at time step t with some latent vector $x_t \in \mathbf{R}^K$ at time step t, and the latent vectors x at earlier time steps. Formally, we define LDS as follows,

$$\begin{aligned} y_t &= C \cdot x_t + v \qquad & x_t &= A_t \cdot x_{t-1} + w \\ v &\sim \mathcal{N}(0, R) \qquad & w &\sim \mathcal{N}(0, Q) \end{aligned}$$

where $C \in \mathbf{R}^{M \times K}$ is the item-factor matrix, and $A_t \in \mathbf{R}^{K \times K}$ is the factor to factor mapping between adjacent time steps. The covariance matrices $Q \in \mathbf{R}^{K \times K}$ and $R \in \mathbf{R}^{M \times M}$ are set to be $0.1 \cdot \mathbf{I}$ in our experiments.

The above LDS formulation models only a single user's data across time steps. It can be extended to model dynamic data of a set of users in a dynamic matrix factorization model. Sun et al. defined a version of DMF as follows [99]:

$$\begin{aligned} y_{n,t} &= C_n \cdot x_{n,t} + v \qquad & x_{n,t} &= A_{n,t} \cdot x_{n,t-1} + w \\ v &\sim \mathcal{N}(0, R) \qquad & w &\sim \mathcal{N}(0, Q) \end{aligned}$$

This version of DMF learns a fixed item-factor matrix C for all users. Instead of learning C, we propose to use an item-factor matrix derived from NMF. We also propose four other versions of DMF based on the options used for **Item Factor Matrix** and **Dynamics Matrix** as shown in Table 5.1.

Basic DMF (DMF-B). In this Basic DMF model, we determine a static

Table 5.1: Proposed DMF Models

	Non-Scaled Item Factors	Scaled Item Factors
Variable Dynamics Matrix	DMF-B	DMF-I
Fixed Dynamics Matrix	DMF-A	DMF-IA

item-factor matrix C using NMF while allowing the factor-user matrix X_t to vary with time. That is, we define Basic DMF to be:

$$y_{n,t} = C \cdot x_{n,t} + v \quad Y^* = C \cdot X^* \quad \text{using NMF}$$

keeping the equations of $x_{n,t}$, w and v the same.

To use DMF for obtaining an estimate of $\hat{y}_{m,n,t}$, we calculate $y_{n,m,t|T}$, the frequency of adoptions by user n on item m at time t conditioned on all information up to the last time step T.

$$y_{n,t|T} = C \cdot x_{n,t|T}$$

DMFs with Scaled Item Factors (DMF-I and DMF-IA). The DMF-I and DMF-IA models consider that the item-factor matrix C learnt from NMF is determined for the observations for *all* time steps, i.e., Y^*. With large observed adoption counts in Y^*, we expect larger entries in C. The consequence of this is an over-estimation of item factors for each time step. Consider using the NMF model to recover adoptions, we have

$$y^*_{m,n} = \sum_{k=1}^{K} c_{m,k} \cdot x^*_{k,n}$$

However, in DMF, we have

$$y_{m,n,t} = \sum_{k=1}^{K} c_{m,k} \cdot x_{n,t,k}$$

In NMF, both C and X^* contribute to the observation of the magnitude in

Y^* for the respective indices. However, in DMF-B, the adoption magnitude Y is spread out over multiple time periods. If C remains constant when inferring for the values of $x_{n,t}$, the value of $x_{n,t}$ will have to be adjusted downwards in order to compensate for the reduction of the observed value $y_{m,n,t}$. While it is convenient to allow $x_{n,t}$ to bear the burden of adjusting for $y_{m,n,t}$, we could also adjust C such that it is suitable for the number of observed time steps for each user n. For example, if a user is only active in one time step, then C_n should be no different with the C from NMF. However, if user is active in multiple time steps, then C_n for user n should be scaled such that $C_n < C$. In the DMF-I model, we therefore scale C by the number of time steps.

$$C_n = \frac{C}{\#\text{ of observed time steps for user } n}$$

Alternatively, C can be estimated via a log likelihood maximization approach in the same way as how A is optimized. But the elegance of how LDS is being defined allows for the parallel estimation of the x_n's and A_n's parameters independently from each user n. Therefore if we learn the C that is coupled with all other users, it becomes computationally expensive with little room for parallelization and scalability.

DMFs with Fixed Dynamics (DMF-A and DMF-IA). In both DMF-B and DMF-I, the dynamics matrix A is different (or variable) for each user and each time step. In predicting missing total adoptions, we have user-item pairs that do not involve any adoption across all time steps. Using different dynamics matrices across time steps may cause over-fitting problem in DMF-B and DMF-I and prevent accurate prediction of missing total adoptions. We therefore propose to learn a fixed dynamics matrix A for each user across all time steps. DMF-A and DMF-IA thus have the following equation for $x_{n,t}$.

$$x_{n,t} = A \cdot x_{n,t-1} + w$$

Parameter Learning for DMF. The estimation of parameters in all the DMF models can be derived as laid out in Rauch, Tung and Striebel [82] and that of Ghahramani and Hinton [40]. In the following, we only show the learning of parameters for DMF-B and DMF-I.

Let $x_{n,t|T}$ be the *smoothed* latent state variable of user n at time t conditioned on T. Without showing the explicit derivations, we only state the equations here. Readers interested in the derivations can refer to Rauch, Tung and Striebel [82]. The steps listed here is known as *RTS smoothing*.

$$
\begin{aligned}
x_{n,t|T} &= x_{n,t|t} + J_{n,t}\left(x_{n,t+1|T} - x_{n,t+1|t}\right) \\
J_{n,t} &= P_{n,t|t}A'_{n,t}P^{-1}_{n,t+1|t} \\
P_{n,t|T} &= P_{n,t|t} + J_{n,t}\left(P_{n,t+1|T} - P_{n,t+1|t}\right)J'_{n,t}
\end{aligned}
$$

Kalman The smoothed latent states depends on the prior latent states $x_{n,t|t-1}$ and posterior latent states $x_{n,t|t}$. The posterior and prior latent states are obtained through a process known as *Kalman filtering* [49].

$$
\begin{aligned}
x_{n,t|t-1} &= A_{n,t}x_{n,t-1|t-1} \\
P_{n,t|t-1} &= A_{n,t}P_{n,t-1|t-1}A'_{n,t} + Q \\
K_{n,t} &= P_{n,t|t-1}C'\left(CP_{n,t|t-1}C' + R\right)^{-1} \\
x_{n,t|t} &= x_{n,t|t-1} + K_{n,t}\left(y_{n,t} - Cx_{n,t|t-1}\right) \\
P_{n,t|t} &= \left(I - K_{n,t}C\right)P_{n,t|t-1}
\end{aligned}
$$

The dynamics matrix $A_{n,t}$ is given by

$$
A_{n,t} = \left(x_{n,t|T} \cdot x'_{n,t-1|T} + P_{n,t,t-1|T}\right)\left(x_{t-1|T} \cdot x'_{t-1|T} + P_{n,t-1|T}\right)^{-1}
$$

Although the matrix C remains the same as before, the latent space vectors $x_{n,t}$, now divided by different time steps no longer have their non-negativity

constraints enforced by the Kalman filtering and smoothing steps. This is because when solving for the posterior and smooth distributions of x, it also involves a maximization step without additional constraints on the polarity of the vectors. Although Lagrange constraints can be added to enforce x to lie on the positive orthant, the algebraic manipulations becomes far too complex to solve analytically. Stochastic gradient descent can be used for solving the posterior and smoothed vectors numerically but given the multiple time steps involved for multiple users, the complexity of such an approach is not feasible for data on a larger scale.

5.3 Experiments

We evaluate our models against the baseline NMF and TimeSVD++[1] on the three tasks: 1) DMF-B and DMF-I for *Prediction of missing temporal adoptions*, 2) DMF-B and DMF-I for *Prediction of all total adoptions given missing temporal adoptions* and 3) DMF-A and DMF-IA for *Prediction of missing total adoptions*. Evaluations on tasks 1 and 2 use the same training and testing sets while task 3 uses a different training and testing sets. In this section, we will discuss how the training and testing sets are constructed before reporting the results for the three tasks.

5.3.1 Data Set

We use a subset of publications from DBLP and ACM Digital Library (ACMDL). Using papers published in the Journal of ACM (JACM) as a seed set, we grow this seed set by including their authors and their non-JACM publications. We also include the co-authors of JACM authors, and the publications of these co-authors. We collect the titles and abstracts (for ACMDL only) of all the above publications.

[1]The TimeSVD++ we used is the implementation from GraphLab

The statistics of our data sets are given in Table 5.2. In this experiment, we use authors and title/abstract words as users and items respectively. Each year is considered a time step. DBLP has twice as many authors as ACMDL due to the longer history of publications maintained by DBLP. DBLP covers a larger scope than ACMDL as the latter focuses only on ACM-related publications. However, ACMDL has many more unique words than DBLP, because ACMDL has both titles and abstracts, whereas DBLP only has titles.

Table 5.2: Dataset Sizes

Data set	# authors	# unique non-stop words	# non-zero entries in Y	time steps
DBLP	52,754	20,080	4,085,265	1936–2012
ACMDL	24,569	33,044	8,721,385	1952–2011

Training and Testing sets for Task 1: To evaluate Task 1 (Prediction of missing temporal adoptions), we divide the temporal adoption matrix Y into five (training set, testing set) pairs, $(Y(i)^{train}, Y(i)^{test})$ for i=1 to 5. The process for creating these data sets is outlined as follows,

1. $Y(0)^{train} = Y$, $Y(0)^{test} = \emptyset$
2. For i=1 to 5
   (a) $Y(i)^{train} = Y(i-1)^{train}$
   $Y(i)^{test} = Y(i-1)^{test}$
   (b) For each $y(i)^{train}_{m,n,t} > 0$, with probability 0.1, do
      i. $y(i)^{test}_{m,n,t} = y(i)^{train}_{m,n,t}$
      ii. $y(i)^{train}_{m,n,t} = 0$

We deliberately hide 10% of adopted items in each time step. We then iteratively grow the testing set by shifting 10% of the adoptions in the training set to the testing set. This way, we can ensure that subsequent testing set is always a superset of the previous set. That makes the difficulty of predicting for the missing adoptions in the test set consistently more difficult than the

previous set. We obtain five sets of testing data { 10%, 19%, 27%, 34%, 41% } with their respective training data.

Training and Testing sets for Task 2: For Task 2 Prediction of all total adoptions given missing temporal adoptions, we simply collapse the above $Y(i)^{train}$ and $Y(i)^{test}$ across time steps. That is, for each i, the training and test sets are defined by,

$$y^*(i)^{train}_{m,n} = \sum_t y(i)^{train}_{m,n,t} \qquad y^*(i)^{test}_{m,n} = \sum_t y(i)^{test}_{m,n,t}$$

Training and Testing sets for Task 3: For task 3, we divide the temporal adoption matrix Y into training sets $Y(j)^{train}$ and testing sets $Y(j)^{test}$, for j=1 to 5. The process for creating these data sets is listed as follows,

1. $Y(0)^{train} = Y$, $Y(0)^{test} = \emptyset$
2. $Y^*(0)^{train} = Y^*$, $Y^*(0)^{test} = \emptyset$
3. For j=1 to 5
   (a) $Y(j)^{train} = Y(j-1)^{train}$
   $Y(j)^{test} = Y(j-1)^{test}$
   (b) $Y^*(j)^{train} = Y^*(j-1)^{train}$
   $Y^*(j)^{test} = Y^*(j-1)^{test}$
   (c) For each $y^*(j)^{train}_{m,n} > 0$, with probability 0.1, do
      i. $y^*(j)^{test}_{m,n} = y^*(j)^{train}_{m,n}$
      $y^*(j)^{train}_{m,n} = 0$
      ii. For t=1 to T
         A. $y(j)^{test}_{m,n,t} = y(j)^{train}_{m,n,t}$
         B. $y(j)^{train}_{m,n,t} = 0$

We create the testing set by randomly including 10% of the item m-user n pairs with non-zero $y^*_{m,n}$ from Y^*. The selected pairs are also excluded from

the training set by setting $y_{m,n,t} = 0$ for all t. The size of the training and testing sets is then varied by randomly selecting another 10% from the training set and shifting it to the testing set.

5.3.2 Results for Prediction of Missing Temporal Adoptions

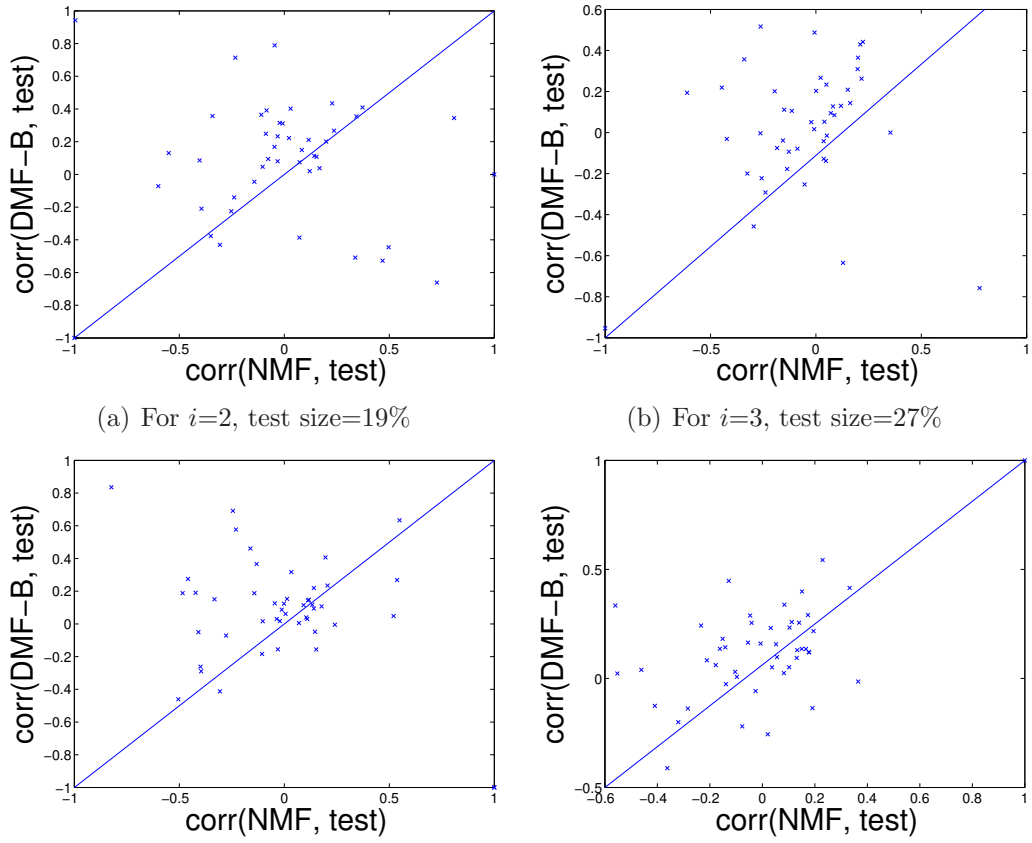
We used $Y(i)^{train}$ for training DMF-B/DMF-I and per time step data from $Y(i)^{train}$ for training NMF. The models then predict the missing adoptions for each time step $y(i)^{test}_{m,n,t}$ for all $y(i)^{test}_{m,n,t} > 0$. The purpose of this experiment is to show that even when NMF is applied independently to each time step, DMF-B and DMF-I are still able to outperform NMF. This indicates that the relationship between the user latent factors of adjacent time steps $x_{n,t}$ and $x_{n,t-1}$ captured by the dynamics matrix is necessary to more accurately predict missing temporal adoptions.

The predicted values given by NMF, DMF-B and DMF-I are denoted by $\hat{y}(i)^{nmf}_{m,n,t}, \hat{y}(i)^{dmf-b}_{m,n,t}$ and $\hat{y}(i)^{dmf-i}_{m,n,t}$ respectively. We compare the *Pearson Correlation Coefficient* (PCC) of the predicted values for each time step against $y(i)^{test}_{m,n,t}$ of each time step. PCC is preferred over *Root Sum Squared Error* (RSSE)[2] because the total adoptions when divided into multiple time steps have many small count values dominating RSSE over the large count values that are deemed more important.

Figure 5.2 shows the result of DMF-B against the baseline NMF using ACMDL dataset for different proportions of test data. In the plot, The x-axis represents the PCC of NMF predicted values against the test (or ground truth) values while y-axis represents the PCC of DMF- predicted values against the test values. Each dot represents the respective results of a year. If the dot lies on the upper-left side of graph, it indicates that for that year DMF-B performs better than NMF. Figure 5.2 indeed shows that for the four plots, most of the

[2]Root Sum Squared Error is defined by the root of squared errors, i.e., $\sqrt{\sum_k error_k^2}$.

dots lie on the upper left side of the figure.



(a) For i=2, test size=19%

(b) For i=3, test size=27%

(c) For i=4, test size=34%

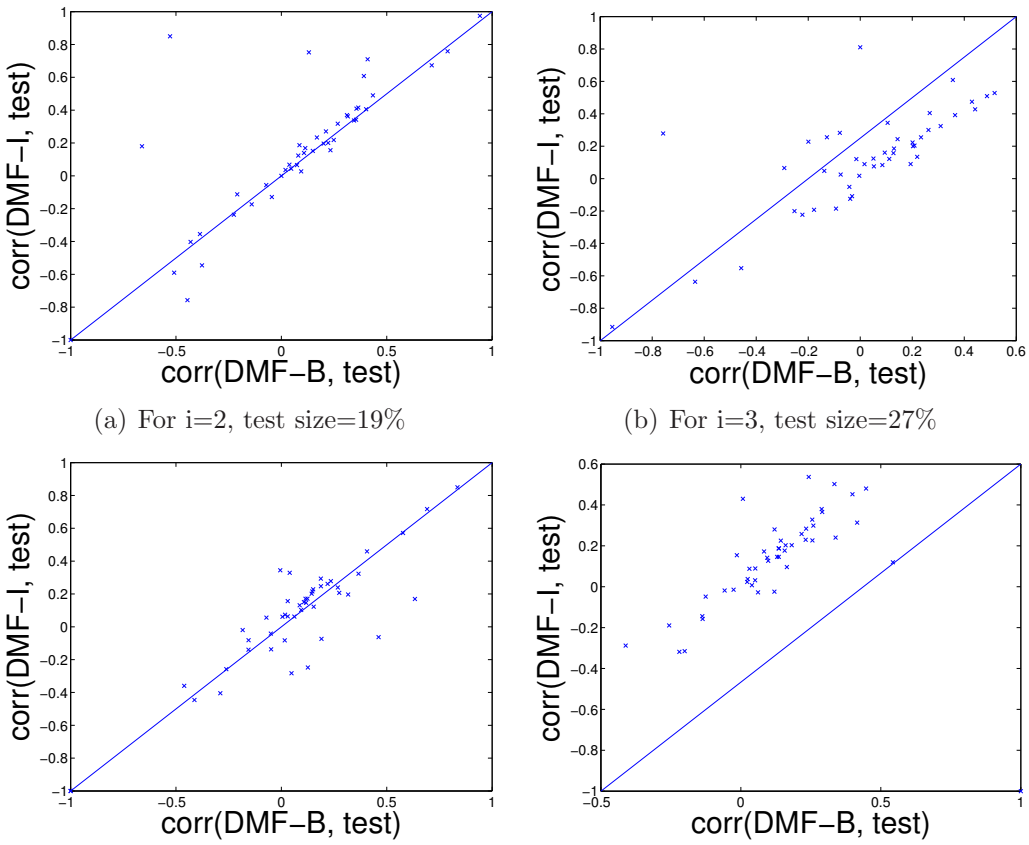
(d) For i=5, test size=41%

Figure 5.2: PCC of DMF-B against NMF for Task 1 (ACMDL)

Figure 5.3 shows the results of DMF-I against DMF-B. The results show that most of the dots lie on the upper left side of the figures. This indicates that using a scaled item-factor matrix C achieve a better estimation of the latent factors. The two figures show that for most years, DMF-I outperforms DMF-B while DMF-B outperforms NMF. Due to space constraints and the small adoption values in each time step, we do not include DBLP for this task.

5.3.3 Results for Prediction of Total Adoptions with Missing Temporal Adoptions

In this evaluation task, the training of DMF-B and DMF-I uses temporal adoptions in $Y(i)^{train}$ while training of NMF uses total adoptions in $Y^*(i)^{train}$. We evaluate how accurate these models predict the total adoptions in $y^*_{m,n}$ for



(a) For i=2, test size=19%

(b) For i=3, test size=27%

(c) For i=4, test size=34%

(d) For i=5, test size=41%

Figure 5.3: PCC of DMF-I against DMF-B for Task 1 (ACMDL)

all $y^*(i)^{test}_{m,n} > 0$ where

$$y^*_{m,n} = \sum_{t=1}^{T} y_{m,n,t}$$

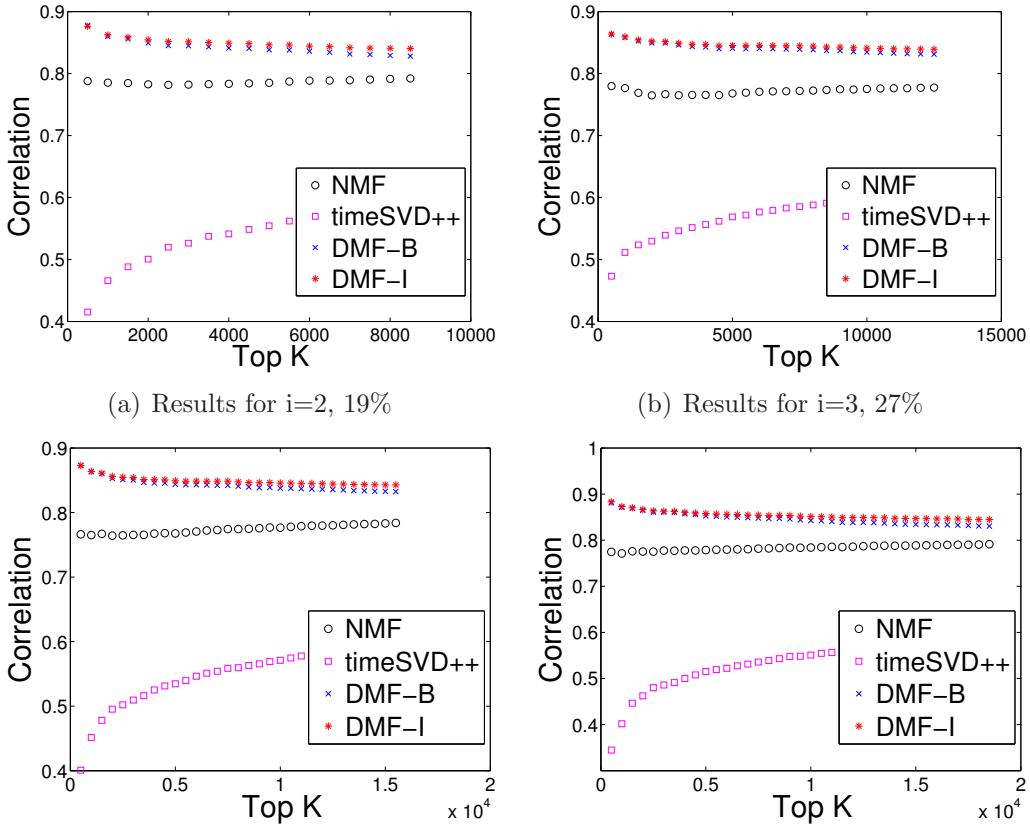
$$y_{m,n,t} = y(i)^{train}_{m,n,t} + y(i)^{test}_{m,n,t}, \text{ for all } i = 1 \text{ to } 5$$

Using DMF-B or DMF-I, we can compute the value of $\hat{y}_{m,n,t}$ for each different time step t. Then an estimate of $\hat{y}^*_{m,n}$ is obtained by summing the predicted value across all time steps.

$$\hat{y}^*_{m,n} = max(\sum_{t=1}^{T} \hat{y}_{m,n,t}, 0)$$

If $\hat{y}^*_{m,n}$ is negative, it is unlikely user m adopts item n and we set the predicted adoption value to zero.

We evaluate the predicted adoption values against the test (ground truth) values using Pearson correlation coefficient (PCC) and Root Sum Squared Error (RSSE) for k largest test adoption values where k is varied from 1 to the number of test cases with adoption values not smaller than 20, ignoring the less important small adoption values.



(a) Results for i=2, 19%

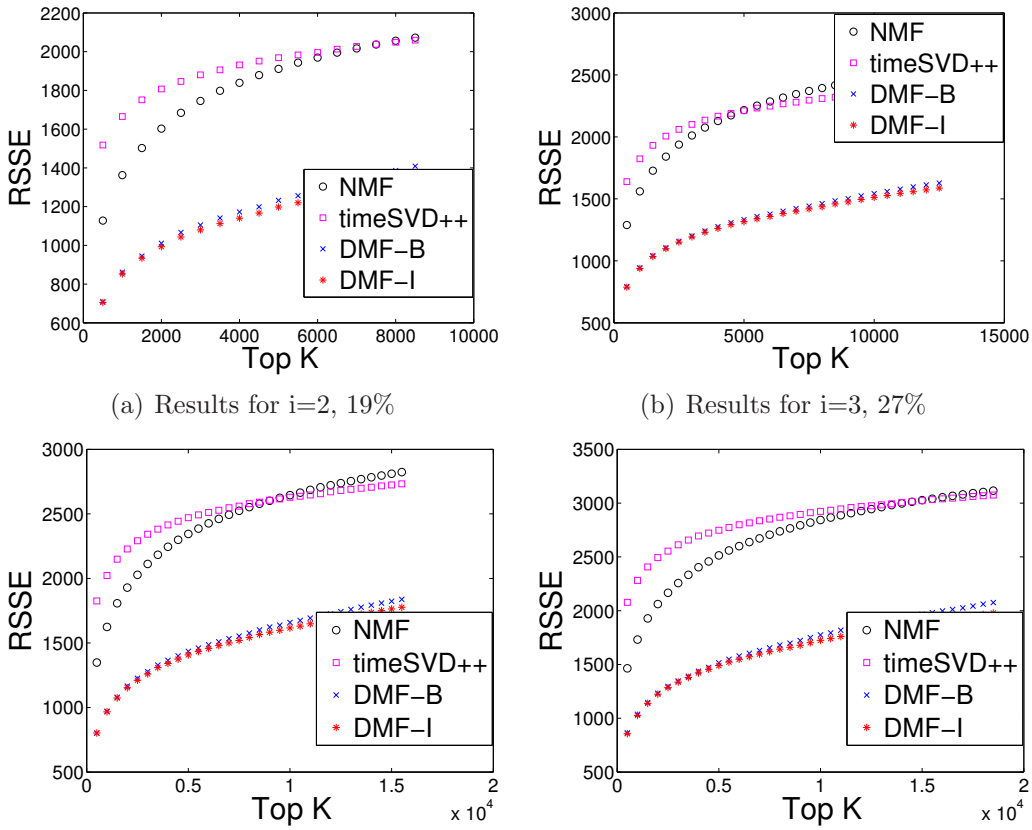
(b) Results for i=3, 27%

(c) Results for i=4, 34%

(d) Results for i=5, 41%

Figure 5.4: PCC of Task 2 (ACMDL)

Figures 5.4 and 5.5 show the correlation and RSSE results for Task 2. The results show that DMF-I outperforms DMF-B by a very small margin and the DMF-B and DMF-I outperforms NMF and TimeSVD++ by a large margin. This indicates that the two DMF models can recover the total adoptions more accurately when some temporal adoptions are missing. The PCC and RSSE performance reduces as we increase k adding more errors to the measures. We also perform similar experiments on the DBLP data set. As shown in Figures 5.6 and 5.7, we also observe that DMF-B and DMF-I outperforms NMF and TimeSVD++ significantly by PCC and RSSE.



(a) Results for i=2, 19%

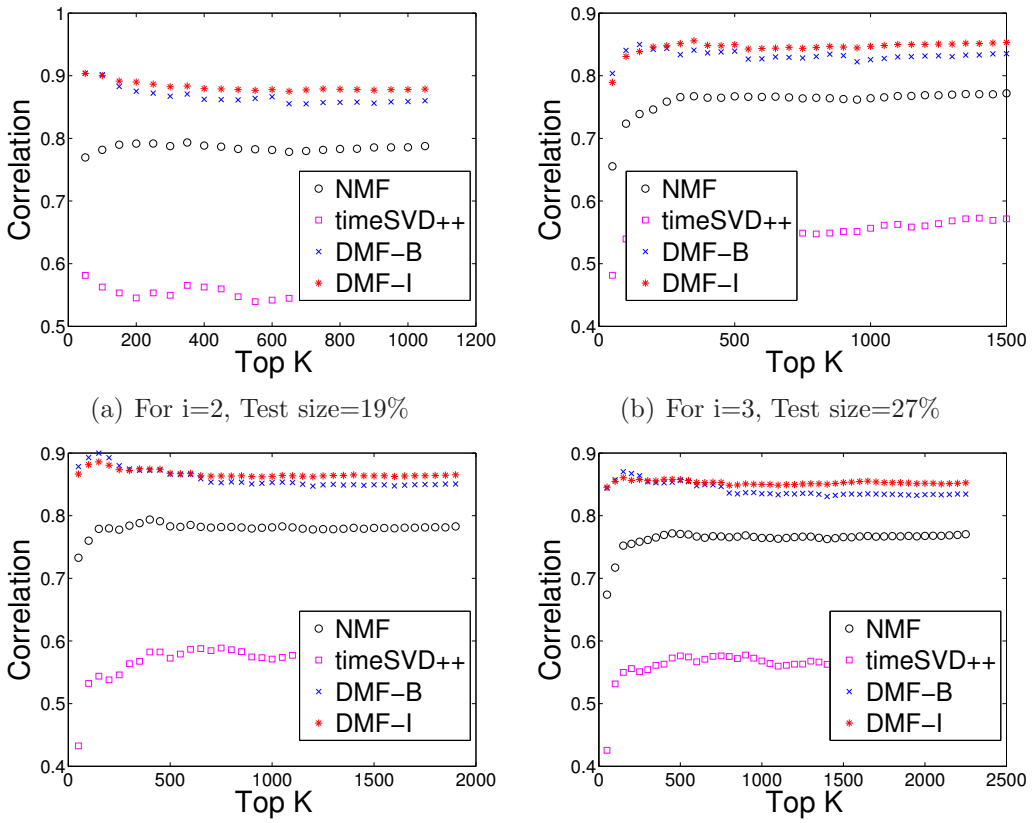
(b) Results for i=3, 27%

(c) Results for i=4, 34%

(d) Results for i=5, 41%

Figure 5.5: RSSE of Task 2 (ACMDL)

While it is expected that NMF will perform poorly on task 2 due to the lack of temporal considerations, we are surprised that TimeSVD++ also performs as poor as NMF on task 2. Manual inspection of the predicted values given by TimeSVD++ shows that TimeSVD++ predicts almost the same adoption values $\hat{y}^*_{m,n,t}$ for all time steps t where user n is active in. Given that for task 2, user adoption values for an item m is missing in some but not all of the time steps, an adoption model should cope with such variations in item user adoption values throughout the entire temporal duration. Since TimeSVD++ was originally developed for user-item rating prediction, it assumes that once item has been rated by user, the rating remains the same throughout the entire temporal duration. Such an assumption violates the conditions necessary for good prediction in task 2.



(a) For i=2, Test size=19%

(b) For i=3, Test size=27%

(c) For i=4, Test size=34%

(d) For i=5, Test size=41%

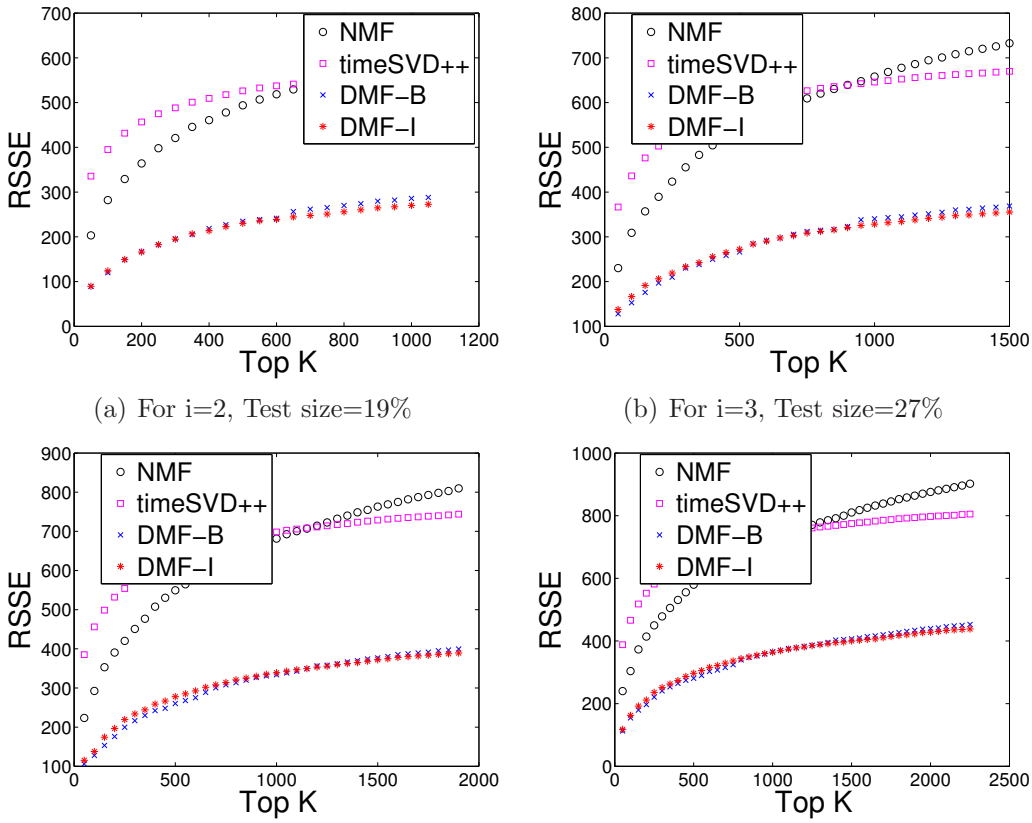
Figure 5.6: PCC of Task 2 (DBLP)

5.3.4 Results for Missing Total Adoptions

For this task, we train DMF-A and DMF-IA using $Y(j)^{train}$ and train NMF using $Y^*(j)^{train}$. We want to investigate if the temporal adoption data of known user-item pairs can help to predict the missing total adoption for a given user-item pair. The test total adoptions to be predicted are $y^{*,test}_{m,n}$ for all $Y^*(j)^{test} > 0$ where

$$y^{*,test}_{m,n} = \sum_{t=1}^{T} y^{test}_{m,n,t}$$

DMF-A and DMF-IA are required to predict the values of $\hat{y}_{m,n,t}$ for each different time steps t. They then give an estimate of $\hat{y}^*_{m,n}$ by summing the



(a) For i=2, Test size=19%

(b) For i=3, Test size=27%

(c) For i=4, Test size=34%

(d) For i=5, Test size=41%

Figure 5.7: RSSE of Task 2 (DBLP)

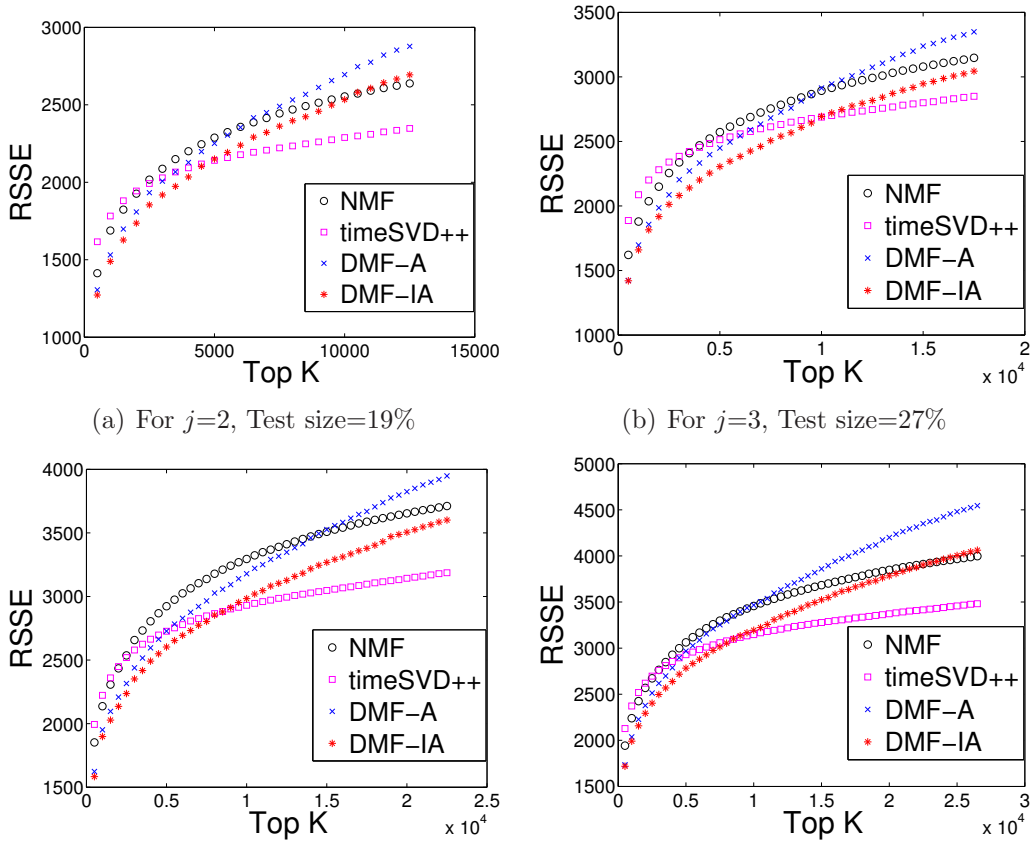
predicted value across all time steps.

$$\hat{y}^*_{m,n} = max(\sum_{t=1}^{T} \hat{y}_{m,n,t}, 0)$$

If $\hat{y}^*_{m,n}$ is negative, we set it to zero. The predicted total adoptions are then compared with test (ground truth) adoptions $y^*_{m,n}$ by Root Sum Squared Error (RSSE).

We again evaluate the predicted adoption values against the test (ground truth) values using PCC and Root Sum Squared Error RSSE for k largest test adoption values where k is varied from 1 to the number of test cases with adoption values not smaller than 20, ignoring the less important small adoption values. Figures 5.8 and 5.9 shows the RSSE results for the ACMDL and DBLP data set. DMF-IA is observed to have smaller RSSE than DMF-I showing that fixed dynamics matrix and scaled item factors are required to yield

more accurate predictions than DMF-I and NMF for this task. DMF-I again outperforms NMF for PCC and RSSE predictions. However, TimeSVD++ have better performance for task 3 in the comparison of RSSE values. Since for task 3, the adoption values of item m and user n are consistently missing for all time steps, the adoption model does not have to make different prediction values for different time steps. When comparing against the aggregated item adoption values $Y^*(i)$, this hides the weakness of rating prediction models such as TimeSVD++.



(a) For j=2, Test size=19%

(b) For j=3, Test size=27%

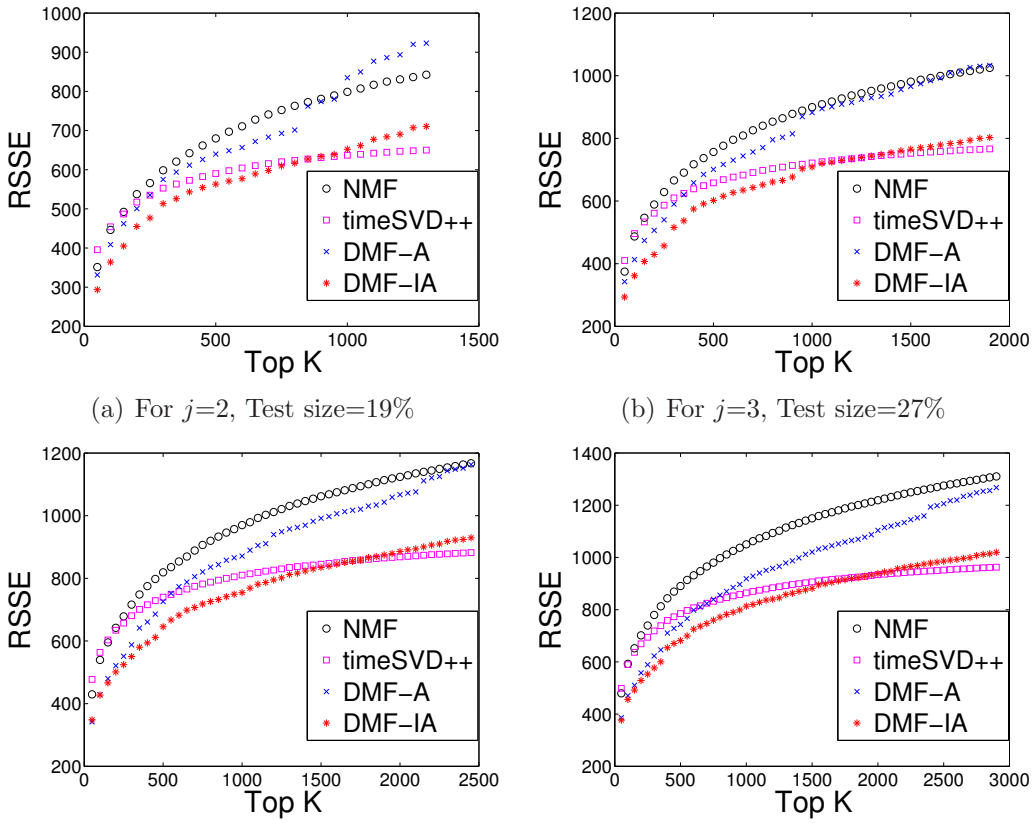
(c) For j=4, Test size=34%

(d) For j=5, Test size=41%

Figure 5.8: RSSE of Task 3 (ACMDL)

5.3.5 Case Study

A main feature of DMF formulation is the use of dynamics matrix $A_{n,t}$ to capture the evolution of user n's latent factors $x_{n,t}$ from one time step to the



(a) For j=2, Test size=19%

(b) For j=3, Test size=27%

(c) For j=4, Test size=34%

(d) For j=5, Test size=41%

Figure 5.9: RSSE of Task 3 (DBLP)

next time step. The latent state at t is given by

$$x_{n,t} = A_{n,t} \cdot x_{n,t-1}$$

The k^{th} factor in $x_{n,t}$ is derive by the dot product of the k^{th} row of $A_{n,t}$ and $x_{n,t-1}$. The largest value in the k^{th} row of $A_{n,t}$, say the (k, l) value, tells us that the l^{th} latent factor in $x_{n,t-1}$ plays a significant role in explaining for the value of the k^{th} latent factor in $x_{n,t}$.

Using the same author *Duminda Wijesekera* as given in Chua et al. [28], we explain the evolution of *Duminda Wijesekera*'s latent factors for the years (2000 to 2001) and (2001 to 2002). From the item factor matrix C, we can derive the underlying topics of some latent factors as shown in Table 5.3. *Duminda Wijesekera* has research interests in security, multimedia, networks, etc.. His 6^{th} latent factor, corresponding to security topic, evolves from 2.25

in 2000 to 3.23 in 2001, and later to 9.10 in 2002. We also notice that the $(6, 20)^{th}$ entry in the 6^{th} row of $A_{Duminda,2001}$ has the highest value of 0.347 while the other entries in the same row have a mean value of 0.0387. In addition, in the 6^{th} row of $A_{n,2002}$, the $(6, 6)^{th}$ entry has the highest value of 0.3625 while the other values have mean value of 0.1418. This suggests that *Duminda Wijesekera* shifted his research from databases to security from 2000 to 2001. Then from 2001 onwards, the security topic continues to be his main research topic.

Consider another well known author *Christos Faloutsos* who has published widely in databases, data mining and graph mining. The 23^{th} factor of *Christos Faloutsos*, corresponding to graph mining, increased from 2.90 in year 2006 to 14.83 in year 2007. By inspecting his dynamics matrix $A_{Christos,2007}$, we noticed that the $(23, 20)^{th}$ entry of the 23^{th} row has the highest value of 0.5055 while the mean value of other entries in the same row is 0.1056. This indicates that *Christos Faloutsos*'s increased research in the graph mining comes from his previous research interest in databases.

Table 5.3: Latent Factors

Factor 6	Factor 20	Factor 23	Factor 18
access	data	mining	network
control	large	graph	networks
paper	database	cache	wireless
systems	approach	graphs	nodes
based	techniques	frequent	sensor
model	algorithms	patterns	traffic
information	efficient	memory	infiniband
security	stream	vertices	routing
system	query	pattern	mobile
policies	problem	vertex	node

Finally, we observe that another database researcher *Beng Chin Ooi* has shifted his research interests from database to mobile systems between the years 2003 to 2004. The 18^{th} factor (corresponding to mobile systems) of his latent state increased from 1.6907 to 9.1483 between 2003 and 2004. In the 18^{th} row of *Beng Chin Ooi*'s dynamics matrix $A_{Beng\ Chin,2004}$, the $(18, 20)^{th}$

entry shows a large magnitude of 0.2441 while the rest of the other factors give a mean value of 0.0933. This indicates that the increase in mobile systems came from previous involvement with databases.

We stress again that without the use of NMF for DMF, we will not be able to observe such case studies for individual authors.

5.4 Summary

We have highlighted the differences between rating prediction and adoption prediction. When the data given contains temporal information, we proposed the use of Dynamic Matrix Factorization (DMF) for modeling the dynamics of latent states for every user. The empirical results show that using DMF gives overall better performance over NMF and state of the art method such as TimeSVD++. Our case study shows three examples of well-known researchers who changed the focus of their research career from a particular field to other fields in Computer Science. By analyzing the different latent states at different time steps, we can notice the years which indicate a tipping point in their focus. Then by further analyzing the dynamics matrix for the tipping point years, we can observe which fields they contributed to the interest in their respective new fields. Without the non-negative constraints in the item factor matrix for DMF, we will not be able to obtain latent factors that can be interpreted as topics of interests for the users. Therefore, the models proposed here can be used as a form of dynamic topic models for tracking the evolution of users' behavior over time. Temporal data sets have also been gaining attention [38] and the models we highlighted here could be applied to other social media data sets as well.

Chapter 6

Using Linear Dynamical Topic Model for Granger Causal Temporal Social Correlation

The abundance of online user data has led to a surge of interests in understanding the dynamics of social relationship using computational techniques. To this end, we propose to model user's item adoption data for measuring the inter-dependency of adoption behaviors between users over time, termed as *Temporal Social Correlation* (TSC). To address the difficulty of representing users adoption behavior in latent space for sparse adoption data, and estimating the rate of decay in users' preferences over time, we develop a novel dynamic topic model known as Linear Dynamical Topic Model (LDTM). Using the time series constructed from the topic distributions found by LDTM, we then conduct *Granger causality* tests to measure TSC. Extensive experiments on bibliographic data show that the ordering of authors' name plays a statistically significant role on the flow of information between authors. We also present several interesting case studies to highlight the intuition of our analysis.

6.1 Motivation

The proliferation of social media as a widely accepted platform for disseminating ideas and recommending products among connected users motivates research in understanding the relationships among users. Users' relationships is well studied in social science and falls under the area of measuring *social influence*. Measuring social influence has many important applications, such as targeting influential individuals for product marketing, or identifying pivotal people in an organization to optimize corporate management and/or drive innovations.

In the context of social media, we define social influence from a user i to another user j as "the actions of i causes j to perform a set of actions in the future". Social influence has been previously studied by various researchers [33, 34, 70, 100, 26]. However, all of these existing approaches do not take into account the temporal aspects of social influence. We take a step further to consider the temporal dimension in measuring social relationships among users.

Knowing how i's past actions can predict j's future actions better than j's past actions is only a necessary condition and not sufficient for finding social influence. Since the definition of social influence reflects the widely discussed notion of *causality* [45, 79], the sufficient condition for finding social influence requires us to exclude other external factors that could affect the actions of j. That is, we need to eliminate the confounding variables that give doubt to the predictive power of i's and j's past on j's future [44].

However, it is difficult to satisfy this sufficient condition, due to the absence of complete user data capturing all external factors influencing the user's actions. There is also a need to conduct randomized controlled experiments, which is challenging in social networks. Under such constrained scenarios, we relax our assumptions by ignoring the confounding variables and term the simplified notion of social influence as *Granger Causal Temporal Social Corre-*

lation (TSC). We assume that users who are socially correlated tend to make similar choices over time. It is also important to note that the TSC from i to j and that from j to i need not be symmetric.

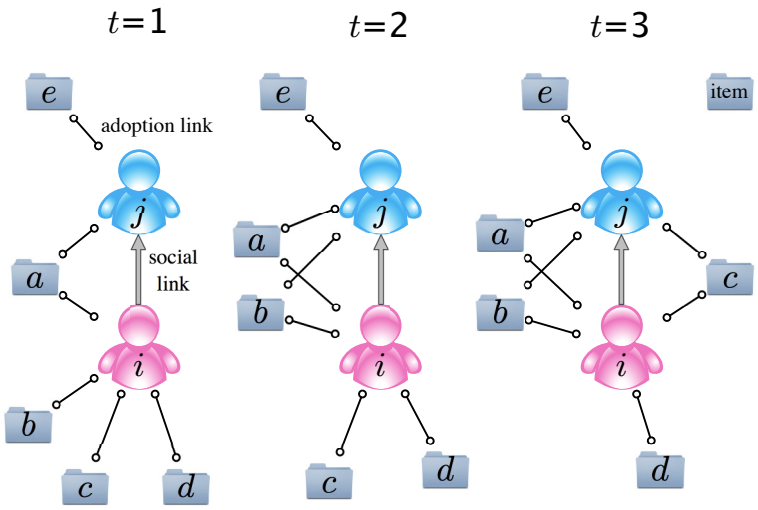


Figure 6.1: Example of temporal social correlation from i to j

We use the term "users adopting items" to describe any action of a user on the media that captures her preferences. Examples of users adopting items include users watching movies, users joining online communities [27, 29] and users producing words [28]. This is also the most common type of user actions in social media, where items may refer to any form of media.

In this chapter, we model the user adoption behavior changes due to temporal social correlation as a form of information transfer between users. We say that "j follows i" leads to information transfer from i to j when there is TSC from i to j at the time point of their interaction τ, denoted as $TSC(i \to j, \tau)$. We also use the term "follow", "information transfer" or "TSC" in place of "social influence", as causality cannot be proven adequately without randomized experiments.

Figure 6.1 shows an example of users i and j adopting different sets of items over three time steps. When temporal information is missing, we could only observe the adoption states at $t = 3$, but it does not tell if i follows j or j follows i. Only by looking at the adoption states of $t = 1$ and $t = 2$, we

can observe that j progressively follows i in adopting items b at $t = 2$ and c at $t = 3$. The converse is unlikely because i's adoption states remain the same over time, that is, i's adoption states at $t = 1$ is sufficient to predict her states for $t > 1$.

Generalizing the example in Figure 6.1, we formulate the following problem: *Given a set of users U and a set of items V that U adopt over time steps 1 to T, determine the $TSC(i \to j, \tau)$ and $TSC(j \to i, \tau)$ for all pair of users $i, j \in U$ when i and j interacts at a specific time point τ.*

A simple yet naive way to quantify $TSC(i \to j, \tau)$ is:

1. To represent the raw frequency of adopted items for i and j at every time step t as a vector $v_{i,t}, v_{j,t} \in \mathbf{R}^M$, where M is the total number of possible items. Vectors for each user i over a set of time steps T form the time series $\{v_{i,1}, \ldots, v_{i,T}\}$.

2. To use the time series $\{v_{i,1}, \ldots, v_{i,T}\}$ and $\{v_{j,1}, \ldots, v_{j,T}\}$ as inputs to existing causality measures such as Granger causality [44] or the more recent *transfer entropy* [102, 103, 104].

However, such approach presents several challenges:

1. The adoption vectors $v_{i,t}, v_{j,t}$ are usually high dimensional in practice (i.e., M is large). In such case, comparing between vectors $v_{i,t}$ and $v_{j,t}$ of two users i and j will be computationally expensive (even using a linear-time algorithm).

2. The vectors $v_{i,t}$ and $v_{j,t}$ are often sparse, since users only adopt a small subset of all possible items. Comparing sparse vectors will hardly yield any indication of significant relationship between them.

3. Since item adoption counts accumulate over time, the change of the adoption vector $v_{j,t}$ relative to its previous time step $v_{j,t-1}$ will become marginal over time, i.e., $\lim_{t\to\infty} \frac{||v_{j,t} - v_{j,t-1}||}{||v_{j,t-1}||} = 0$, where $||.||$ denotes the

Euclidean norm. In this case, the past information of a user j's adoption is sufficient to predict her own future, and the information of other users such as i is no longer useful.

4. If the time series $\{v_{i,1}, \ldots, v_{i,T}\}$ and $\{v_{j,1}, \ldots, v_{j,T}\}$ of users i and j are long, their comparison may give misleading conclusion that no influence exists, because TSC between the two users usually takes place within a window of time periods.

To address 1) and 2), a temporal latent factor model is needed to obtain a dense, compressed representation of adoption behaviors over time. With regard to 3), one may learn the user latent factors at each time step independently. However, (static) adoption data are already sparse, and slicing the data into time steps would further aggravate the problem. To address the sparsity issue, we propose a method to automatically estimate a decay parameter for balancing between the importance of past and recent information. Finally, to handle 4), we specify a time window to constrain the time series comparison period in which social correlation is measured.

To satisfy all these requirements, we propose a novel approach called the *Linear Dynamical Topic Model* (LDTM). The proposed model represents user adoption behavior as topic distributions at different time steps and, for each user, the evolution of the topic distribution parameters is tracked using *Linear Dynamical System* (LDS).

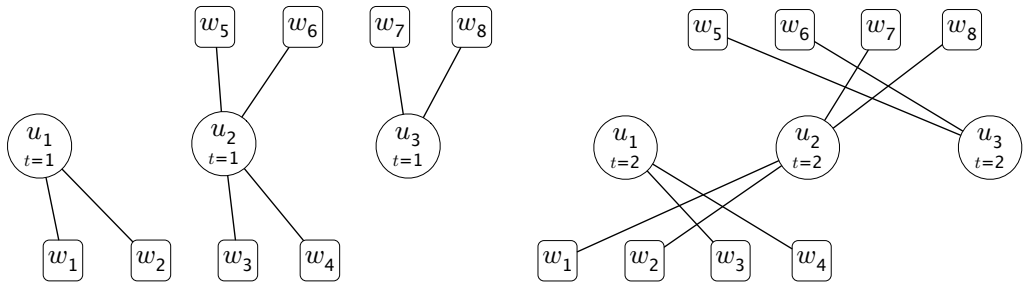
In our experiments, we show that by using the temporal topic distributions for each user, we are able to compare pairs of users by formulating Granger Causality tests. Through experiments on bibliographic data such as DBLP and ACMDL, we find evidence for Granger Causality among the paper co-authors, and our statistical significance tests reveal that the ordering of the co-authors' names plays a role in determining the information transfer among them.

6.2 Issues in Modeling Temporal Adoption Data

Measuring TSC between two users i and j requires two crucial steps. First, an accurate measure of the users' adoption behavior represented as time series vector in latent space is required for every time step. That is, we require latent factor vectors $\theta_{i,t}, \theta_{j,t} \in \mathbf{R}^K$ for each user pair (i, j) at time step t. $\theta_{i,t}$ and $\theta_{j,t}$ has K dimensions and $K << M$. Second, a temporal correlation measure is needed to compare between the trends of two time series. Knowing how two time series temporally correlate should help us make better predictions or reduce our uncertainty for their future adoption behavior. However, we need to address some issues in modeling temporal adoption data, as elaborated in Sections 6.2.1 and 6.2.2.

6.2.1 Modeling Adoption Data Across Time Steps

We propose a new way of representing user's adoption behavior in temporal latent space as opposed to the traditional method of using only the frequency of adoption in high dimensional space. There are some advantages of representing adoption behavior in temporal latent space as well as some difficulties, which we will elaborate further.

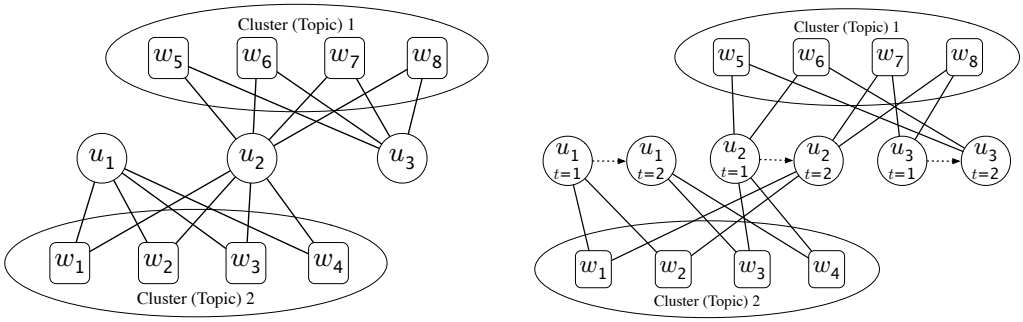


(a) $t = 1$ (b) $t = 2$

Figure 6.2: Topic Modeling in Temporal User Item Adoptions

If we model the topic distributions at each time step independently of other time steps, we would obtain the scenarios in Figures 6.2(a) for time step 1 and 6.2(b) for time step 2. One may see that the edges between users and items

are sparse, which does not allow us to draw any meaningful intuitions about the relationship of items.



(a) Without time steps (b) With time steps

Figure 6.3: Topic Modeling in Static User Item Adoptions

However, when we combine the temporal adoptions into a single time step, we obtain the scenarios as illustrated in Figure 6.3(a). Figure 6.3(a) shows the result of performing topic modeling on data without temporal considerations. The items adopted by users u_1, u_2 and u_3 are clustered according to topics 1 and 2 based on the density of edges between users and items. We therefore require a method of modeling the temporal adoptions such that it allows us to preserve the edge densities across time steps and provides us with the topic distributions at different time steps. Such model could combine the temporal adoptions and construct dependencies between different time steps by having the scenario as shown in Figure 6.3(b).

6.2.2 The Need for Temporal Probabilistic Topic Model

There are many ways of modeling users' adoption behavior in latent spaces, and we wish to justify our choice of using probabilistic topic model. Besides probabilistic method, one may use Non-negative Matrix Factorizations (NMF) to obtain low-rank matrices that can substitute for the users' and items' latent factors [116, 69].

Our previous work in temporal item adoptions has also explored the use of LDS with Non-negative Matrix Factorizations (NMF) [30] for modeling evolv-

ing users' preferences. LDS with NMF can be stated as follows,

$$x_{n,t} = A_{n,t-1} \cdot x_{n,t-1} + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, Q)$$
$$w_{m,n,t} = C_m \cdot x_{n,t}$$

where $x_{n,t} \in \mathbf{R}^K$ is the vector representing users' adoption behavior, $A_{n,t-1} \in \mathbf{R}^{K \times K}$ the dynamics matrix which evolves user's behavior from $t-1$ to t, $w_{m,n,t} \in \mathbf{R}$ represents the number of times user n adopts item m at time t, $C_m \in \mathbf{R}^K$ represents item m's latent factor.

We estimate the items latent factor matrix $C \in R^{M \times K}$ by minimizing the sum-of-squared errors for NMF using Stochastic Gradient Descent (SGD) with non-negative constraints. The model is then solved as an instance of Expectation Maximization (EM) Algorithm [14, 35] by using Kalman Filtering and RTS Smoothing for the E-step and M-step optimizes for the dynamics matrix $A_{n,t}$.

In order to obtain interpretable topics, it is compulsory that the items latent factor matrix contains only non-negative values [30]. This is because we often rank the importance of items according to the items' value in the respective latent factor. But this is not true if the latent factors contain negative values. A negative $c_{m,k}$ can also be important for contributing to the value $w_{m,n,t}$ if the corresponding $x_{n,t,k}$ is also negative.

Due to the different amounts of item adoptions for each user at different time steps, a single static matrix C that is defined in real space $R^{M \times K}$ does not fit well for the adoption patterns of every user. C was also only estimated once before running EM algorithm to estimate the rest of the parameters.

There is thus a strong requirement to have a *non-negative* items' latent factor matrix that is *normalized across different time steps* which is estimated by an algorithm that updates the items' latent factor *iteratively* while learning the other parameters. Probabilistic approaches give us normalized parameters

that sum to one and are non-negative (since probabilities cannot be less than zero). By alternating Gibbs Sampling with Kalman Filter, RTS smoothing and additional maximization steps, we derive an algorithm summarized in Algorithm 2 to estimate all the necessary parameters.

6.3 Linear Dynamical Topic Model

Figure 6.4 shows a graphical representation of LDTM. In essence, LDTM is a combination of Latent Dirichlet Allocation (LDA) and Linear Dynamical System (LDS). We obtain the users' topic distribution at each time step by inferring the latent topic variable conditioned on the words written in each time step and the topic item distribution. We assume that the topic item distribution remains static over time, while the users' topic distribution evolves over time through a linear dynamical process conditioned on the previous time steps and the inferred latent variables in current time step.

6.3.1 Model Assumptions

We describe the assumptions of LDTM as follows:

1. Given that there are K topics and temporal adoption data, the topic distribution $\theta_{n,t}$ of user n at time step t is defined by the Dirichlet distribution with parameters $x_{n,t} \in \mathbf{R}^K$.

$$\theta_{n,t} \sim Dir(x_{n,t})$$

2. To relate the current parameters $x_{n,t}$ with the previous parameters $x_{n,t-1}$,

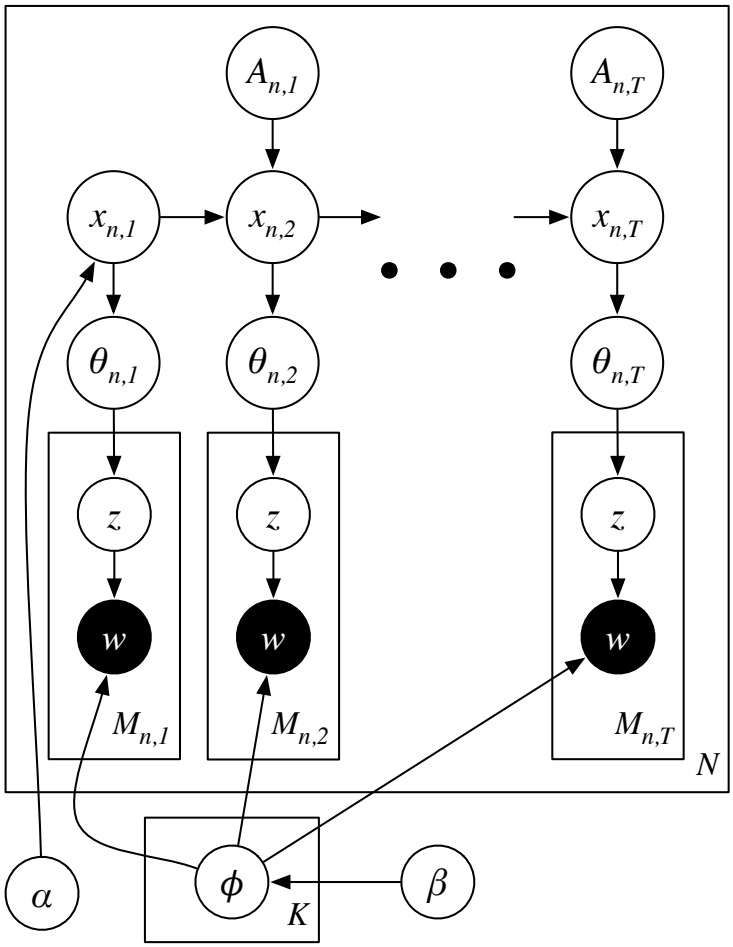


Figure 6.4: Graphical Plate Diagram of LDTM

we assume a linear Gaussian distribution as defined by,

$$x_{n,1} \sim \mathcal{N}(\alpha \cdot \mathbf{1}, Q)$$
$$x_{n,t} = A_{n,t-1} \cdot x_{n,t-1} + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, Q)$$
$$x_{n,t} \sim \mathcal{N}(A_{n,t-1} \cdot x_{n,t-1}, Q)$$

where $A_{n,t} \in \mathbf{R}^{K \times K}$ represents the dynamics matrix of user n at t, and $Q \in \mathbf{R}^{K \times K}$ represents the covariance matrix of the Gaussian noise variable. This step distinguishes our model from all other topic models, i.e., we model the evolution of users' topic distribution using a dynamics matrix. We also derive a whole new set of inference equations for estimating the model parameters in Section B.1.

3. The topic $z_{m,n,t}$ of an item m adopted by user n at time t is given by,

$$z_{m,n,t} \sim Mult(\theta_{n,t})$$

Each topic item distribution is given by a simple symmetric Dirichlet distribution,

$$\phi_k \sim Dir(\beta)$$

Then each item m adopted by user n at time t conditioned on topic variable $z_{m,n,t}$ is given by,

$$w_{m,n,t}|z_{m,n,t} = k \sim Mult(\phi_k)$$

6.3.2 Inference and Parameter Estimation

To calculate TSC, we require the topic distributions for each user n at each time step t conditioned on the information up to t as denoted by $\theta_{n,t|t}$. Since we have defined $\theta_{n,t}$ as a Dirichlet distribution with Gaussian parameters $x_{n,t}$, knowing $x_{n,t|t}$ is sufficient for deriving $\theta_{n,t|t}$. $\theta_{n,t|t}$, which is known as the posterior topic distribution of user n at time t conditioned on information up to time step t is given by,

$$\theta_{n,t|t} \sim Dir(x_{n,t|t})$$

$x_{n,t|t}$, the Gaussian distributed parameters of the Dirichlet distribution for user n at time t conditioned on information up to time step t is given by a slight modification of the **Kalman Filter** [49] algorithm,

$$x_{n,t|t} \sim \mathcal{N}(x_{n,t|t-1} + \psi_{n,t}, Q)$$

where $\psi_{n,t} \in \mathbf{R}^K$ and $\psi_{n,t,k}$ denote the number of times user n at time t generated topic k. $x_{n,t|t-1}$ is the prior distribution of user n at time t conditioned on information up to time step $t-1$,

$$x_{n,t|t-1} = A_{n,t-1} \cdot x_{n,t-1|t-1}$$

where $A_{n,t-1} \in \mathbf{R}^{K \times K}$ is the dynamics matrix that evolves the parameters from $t-1$ to t. If $A_{n,t}$ for all time steps t is assumed to be an identity matrix, the model reduces to the traditional LDA.

6.3.3 Stable Estimation of Decay for Dynamics Matrix

We obtain $A_{n,t}$ by maximizing the log likelihood of the model. The log likelihood portion of the model involving the dynamics matrices is given by,

$$\mathcal{L} = \ldots - \frac{1}{2} \sum_{t=1}^{T-1} \left(x_{n,t+1} - A_{n,t} x_{n,t}\right)' Q^{-1} \left(x_{n,t+1} - A_{n,t} x_{n,t}\right) \ldots$$

$$\frac{\partial \mathcal{L}}{\partial A_{n,t}} = Q^{-1} \left(x_{n,t+1} x'_{n,t} - A_{n,t} x_{n,t} x'_{n,t}\right)$$

By taking the derivative as zero, we obtain,

$$A_{n,t} = \left(x_{n,t+1} x'_{n,t}\right) \left(x_{n,t} x'_{n,t}\right)^{-1}$$

The right-hand side components are given by,

$$x_{n,t+1} x'_{n,t} = V_{n,t+1,t|T} + x_{n,t+1|T} x'_{n,t|T}$$

$$x_{n,t} x'_{n,t} = V_{n,t|T} + x_{n,t|T} x'_{n,t|T}$$

These components $x_{n,t|T}, V_{n,t|T}, V_{n,t+1,t|T}$ require the smoothed parameters conditioned on **all** information from time step 1 to T. In LDS, this is known as **RTS smoothing** [82], which is the continuous analog of the forward-backwards algorithm used in Hidden Markov Models [81]. The combined use of

Kalman Filtering and RTS Smoothing is an instance of Dynamic Programming for Dynamic Optimization [12].

Because we combine the use of RTS smoothing with Gibbs sampling, some set of equations has to be re-derived for use in LDTM. Further details of the derivations are given in Section B.1.

According to Siddiqi et al. [94], an LDS is Lyapunov (aka numerically) stable if the eigenvalues of the dynamics matrix $A_{n,t}$ is less than or equal to one. The eigenvalues of any general matrix are guaranteed to be less than or equals to one if the sum of each row in the matrix is less than or equals to one.

Since our dynamics matrix evolves the parameters of the Dirichlet distribution, the Dirichlet distribution will be invalid if the parameters are negative. In LDTM, there is thus an additional requirement that the entries of the dynamics matrix has to be non-negative. This additional constraint has not been addressed by [94] or any other prior works. We show a simple solution based on the assumption that the dynamics matrix is always diagonal with normalized entries $\in [0, 1]$. This fulfills the stability and non-negativity constraints for the dynamics matrix.

To obtain $A_{n,t} = \left(x_{n,t+1}x'_{n,t}\right)\left(x_{n,t}x'_{n,t}\right)^{-1}$, we assume,

$$G = x_{n,t+1}x'_{n,t}, \qquad H = x_{n,t}x'_{n,t}$$
$$A = GH^{-1}, \qquad A \cdot H = G$$

and we again try to minimize,

$$\sum_{i,k}\left(\sum_{j} A_{i,j} \cdot H_{j,k} - G_{i,k}\right)^2$$

If we assume that the dynamics matrix A is an identity matrix scaled by a

scalar parameter μ, we should try to minimize the following,

$$\begin{aligned} f(\mu) &= \sum_{i,k} (\mu \cdot H_{i,k} - G_{i,k})^2 \\ &= \sum_{i,k} \mu^2 \cdot H_{i,k}^2 - 2\mu H_{i,k} G_{i,k} + G_{i,k}^2 \end{aligned}$$

Taking the derivative with respect to μ

$$\frac{df(\mu)}{d\mu} = \sum_{i,k} 2\mu H_{i,k}^2 - 2H_{i,k}G_{i,k}$$

and then equating it to zero, we get,

$$\mu = \frac{\sum_{i,k} H_{i,k} G_{i,k}}{\sum_{i,k} H_{i,k}^2}$$

Assuming that each diagonal element of A takes a different value μ_i, we get,

$$\mu_i = \frac{\sum_{k} H_{i,k} G_{i,k}}{\sum_{k} H_{i,k}^2} \tag{6.1}$$

After we derive $A_{n,t}$ using (6.1), we scale down (normalize) the values such that $\max\{A_{n,t}\} \leq 1$.

6.3.4 Outline of Parameter Estimation

Algorithm 2 outlines the procedure to estimate the parameters of the model in Figure 6.4. It begins by randomly initializing the latent variables $z_{m,n,t}$, followed by Gibbs Sampling iterations where the distributions are first estimated using Kalman Filter. Then the latent variable $z_{m,n,t}$ is sampled by conditioning on the prior distribution $x_{n,t|t-1}$, previously sampled variables $\psi_{n,t}$ and topic item distribution parameters β. Using the sampled latent variables, we derive the posterior distribution $x_{n,t|t}$ via Kalman Filter, and update the prior and posterior covariances $V_{n,t|t-1}$ and $V_{n,t|t}$ for the later steps of RTS Smoothing. We then perform RTS Smoothing to get the smoothed distributions $x_{n,t|T}$,

gain $J_{n,t}$, smoothed covariance $V_{n,t|T}$ and lag-one covariance smoother $V_{n,t+1,t|T}$. Finally, we estimate the dynamics matrix $A_{n,t}$ and repeat the iterations till convergence.

Algorithm 2 LDTM Inference

Input: Adoption data for each user n at each time step t
Output: Estimated parameters
{Initialization}
for $n \leftarrow 1$ **to** N **do**
  for $t \leftarrow 1$ **to** T_n **do**
    for $m \leftarrow 1$ **to** $M_{n,t}$ **do**
      $k \leftarrow uniformRandom(1, K)$
      {$\psi_{n,t,k}$ denote the number of times user n at time t generated topic k.}
      {$\beta_{k,m}$ denote the number of times topic k generated item m.}
      $\psi_{n,t,k} \leftarrow \psi_{n,t,k} + 1$, $\beta_{k,m} \leftarrow \beta_{k,m} + 1$, $z_{m,n,t} \leftarrow k$
    end for
  end for
end for
{Gibbs Sampling}
while iterate **do**
  for $n \leftarrow 1$ **to** N **do**
    {Kalman Filter}
    for $t \leftarrow 1$ **to** T_n **do**
      $x_{n,t|t-1} \leftarrow A_{n,t-1} \cdot x_{n,t-1|t-1}$
      for $m \leftarrow 1$ **to** $M_{n,t}$ **do**
        $k \leftarrow z_{m,n,t}$, $\psi_{n,t,k} \leftarrow \psi_{n,t,k} - 1$, $\beta_{k,m} \leftarrow \beta_{k,m} - 1$
        $k \leftarrow sample(x_{n,t|t-1} + \psi_{n,t}, \beta_k)$
        $\psi_{n,t,k} \leftarrow \psi_{n,t,k} + 1$, $\beta_{k,m} \leftarrow \beta_{k,m} + 1$, $z_{m,n,t} \leftarrow k$
      end for
      $x_{n,t|t} \leftarrow x_{n,t|t-1} + \psi_{n,t}$
      Update $V_{n,t|t-1}$ and $V_{n,t|t}$
    end for
    {RTS Smoothing}
    for $t \leftarrow T_n$ **to** 1 **do**
      Update $x_{n,t|T}, J_{n,t}, V_{n,t|T}, V_{n,t+1,t|T}$
    end for
    Estimate the dynamics matrix $A_{n,t}$
  end for
end while

6.4 Finding Temporal Granger Causality

After obtaining the posterior topic distributions $\theta_{n,t|t}, \forall n \in U$, we calculate TSC using **Granger causality** [45]. For a pair of users (i, j), TSC can be measured in two directions, $TSC(i \to j, \tau)$ and $TSC(j \to i, \tau)$, pivoted at a specific time step τ. τ is appropriately chosen to indicate the beginning of information transfer between x and y. Given τ, we could then select a time window $[\tau - W, \tau + L]$ to constrain time series used for comparison, where L is the number of time steps to "lookahead" for measuring TSC and W is the "width" of past time steps for predicting the future.

For notational simplicity, we denote the topic distributions for users i and j at t as i_t and j_t respectively. Specifically, given two users i and j who interact at time τ, $TSC(i \to j, \tau)$ is computed as follows:

1. Formulate the two linear regression tasks below:

$$\begin{aligned}
\tilde{j}_t &= \eta_0 + \left(\sum_{w=1}^{W} \eta_w j_{t-w} \right) + \epsilon_1 \\
\epsilon_1 &\sim \mathcal{N}(0, \sigma_1^2) \\
R_1 &= \sum_{t=\tau}^{\tau+L} \left(j_t - \tilde{j}_t \right)^2 \qquad (6.2) \\
\bar{j}_t &= \eta_0 + \left(\sum_{w=1}^{W} \eta_w j_{t-w} + \lambda_w i_{t-w} \right) + \epsilon_2 \\
\epsilon_2 &\sim \mathcal{N}(0, \sigma_2^2) \\
R_2 &= \sum_{t=\tau}^{\tau+L} \left(j_t - \bar{j}_t \right)^2 \qquad (6.3)
\end{aligned}$$

   where τ is the time point when i and j begins transferring information between one another.

2. Estimate for the parameters $\{\eta_0, \ldots, \eta_W\}$ by minimizing the least squares error in (6.2) using **Coordinate Descent** [13], and then estimate *only* for the parameters $\{\lambda_1, \ldots, \lambda_W\}$ by minimizing (6.3). The first linear

regression given by (6.2) uses j's past information to predict j's future, while the second linear regression (6.3) uses additional information from i's past to predict y's future.

3. To obtain the $TSC(i \to j, \tau)$, we measure how much i's past improves the prediction of j's future by computing the F-statistic (*F-stat*),

$$TSC(i \to j, \tau) = \textit{F-stat} = \frac{R_1 - R_2}{R_2} \cdot \frac{2L - 1}{W}$$

Because the formula (6.3) uses more parameters than (6.2), the sum-of-squares error given by R_2 is always smaller than R_1, i.e. $R_2 < R_1$, which implies that *F-stat* is always positive.

4. Repeat the steps for computing $TSC(j \to i, \tau)$ and compare whether $TSC(i \to j, \tau) > TSC(j \to i, \tau)$ or otherwise.

6.5 Experiments

To evaluate the effectiveness of LDTM and the TSC calculated for pairs of users, we require data sets that provide users' temporal adoptions and the interactions between users that lead to information transfer between them. The publicly available DBLP [67] and ACM Digital Library (ACMDL) [1] academic data sets provide the information we require. We first describe how we obtain subsets of the data from DBLP and ACMDL for our evaluation needs. Then we evaluate the effectiveness of LDTM for several scenarios of the dynamics matrix $A_{n,t}$:

1. LDA: To reduce LDTM to the baseline LDA, we simply set $A_{n,t}$ as identity matrix for every user n and every time step t, i.e. $A_{n,t} = I$.

2. Half-Decay: We set $A_{n,t}$ as diagonal matrix with constant values of 0.5, i.e. $A_{n,t} = 0.5 \cdot I$.

3. Full-Decay: We set $A_{n,t}$ as zero matrix, i.e. $A_{n,t} = 0$.

4. LDTM: We automatically determine the values of the dynamics matrix $A_{n,t}$.

We show that automatically estimating $A_{n,t}$ gives us better representations of authors' temporal adoption behavior than setting constant values for $A_{n,t}$. Using the ideal authors' temporal adoption behavior based on LDTM with automatically estimated $A_{n,t}$, we apply the Granger causality tests and obtain the TSC for every pair of interactions between authors. We show that the first author is more likely to follow the proceeding authors in adoption behavior. Finally, we show case studies of several well-known examples to highlight the authenticity of our approach to calculate TSC.

6.5.1 Data Set

We used the DBLP and ACMDL data to obtain our required users and items. The authors who wrote papers are treated as users and the words in their papers are seen as adopted items. We used the words written in the abstract for ACMDL and those written in title for DBLP. The co-authorship information provides a time point where interaction occurred between the two authors.

Given the large number of publications in DBLP and ACMDL, we only use a subset of papers from DBLP and ACMDL. We sample a subset of data that covers a wide variety of fields in Computer Science by using the papers published in the Journal of ACM (JACM) as a seed set. We then expand the coverage by including other non-JACM publications by authors with at least one JACM publication. The sample obtained here is termed *ego-1*. By including the co-authors of the authors in *ego-1* and their publications, we get a larger sample called *ego-2*. We repeat the process once more to obtain *ego-3*.

Table 6.1 gives the sizes of the *ego-2* and *ego-3* data sets we sampled. DBLP has more authors than ACMDL, because DBLP covers a longer his-

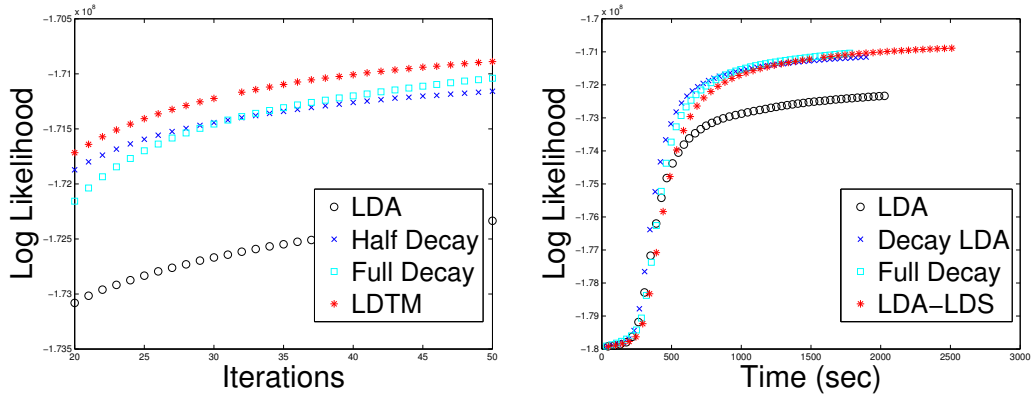
Table 6.1: Data Set Sizes

	#authors	#words	period
ACMDL (ego-2)	24,569	33,044	1952-2011
ACMDL (ego-3)	157,715	44,308	1952-2011
DBLP (ego-2)	52,754	20,080	1936-2013
DBLP (ego-3)	388,092	40,463	1936-2013

tory of publications and has more sources of publications. On the other hand, ACMDL focuses mainly on ACM-related publications. After pruning away the stop-words and non-frequent (less than ten occurrences) words, ACMDL sampled data sets have slightly more words than DBLP, as ACMDL provides words in the abstract of publications while DBLP only has words in the publications title. We use the smaller *ego-2* samples for experiments that require repetitions, and use the significantly larger *ego-3* samples for experiments that only require a single run.

6.5.2 Convergence of Log Likelihood

We first evaluate the convergence of the log likelihood for the case where we automatically estimate the dynamics matrix $A_{n,t}$ and for cases where $A_{n,t}$ is set to constant values. We use the *ego-2* samples for evaluating log likelihood convergence because *ego-2* will be used later for the predictive evaluations.



(a) Log Likelihood vs # of Iterations (b) Log Likelihood vs # Time (seconds)

Figure 6.5: ACMDL: Convergence of Log Likelihood

Figure 6.5(a) shows how log likelihood varies with the number of iterations

(from 20 to 50) for ACMDL (ego-2). We can see that LDTM gives the highest log likelihood, while LDA gives the lowest log likelihood. This suggests that automatically estimating the dynamics matrix gives a better fit to the sampled data as opposed to setting constant values for the dynamics matrix.

Figure 6.5 also reveals another interesting observation: given enough number of iterations, the model with full decay gives better log likelihood than the model with half decay. We attempt to explain this observation as follows. The fit of the estimated parameters depend on the user latent factors and the item latent factors. The introduction of dependencies between temporal adoptions aid the model in estimating better user latent factors only. But given enough number of iterations, the item latent factors converged to a point where the user latent factors become less important. This explains why the log likelihood of full decay outperforms half decay when we increase the number of iterations.

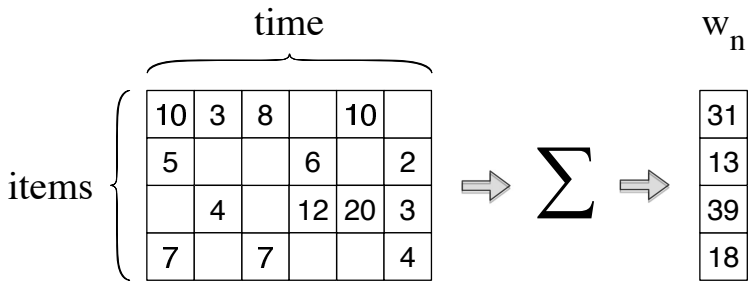
Figure 6.5(b) shows that LDTM takes slightly longer time to complete the same number of iterations as other baselines. We will explain how to improve the efficiency in Section 7.2.

6.5.3 Results for LDTM

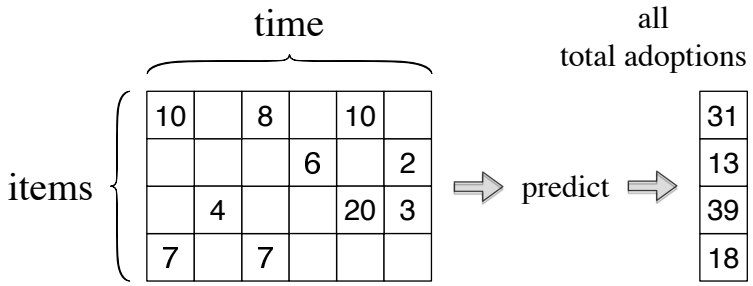
We evaluate the automatic estimation of dynamics matrix $A_{n,t}$ for LDTM by comparing against fixed values of $A_{n,t}$ for two tasks which we described with the aid of Figure 6.6. Figure 6.6(a) shows an example of the original adoption data for an arbitrary user n of four items over six time steps. Each element in the matrix represents the frequency of adoption for the respective item (row) in that time step (column). The aggregation over all time steps gives us the *total adoption* of the user n as represented by w_n in Figure 6.6(a).

The objective of the two tasks depends on how we derive the training sets which we describe as follows:

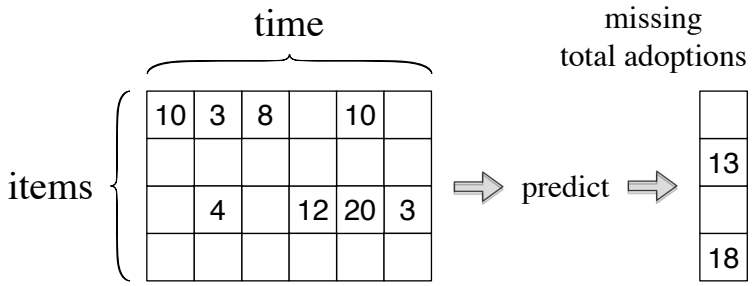
1. **Task 1:** For each user in the original data, we pick at random an (item, time) pair and hide it in the training data set. Figure 6.6(b) shows the



(a) Original Data Set for a User



(b) Training Data Set for Task 1



(c) Training Data Set for Task 2

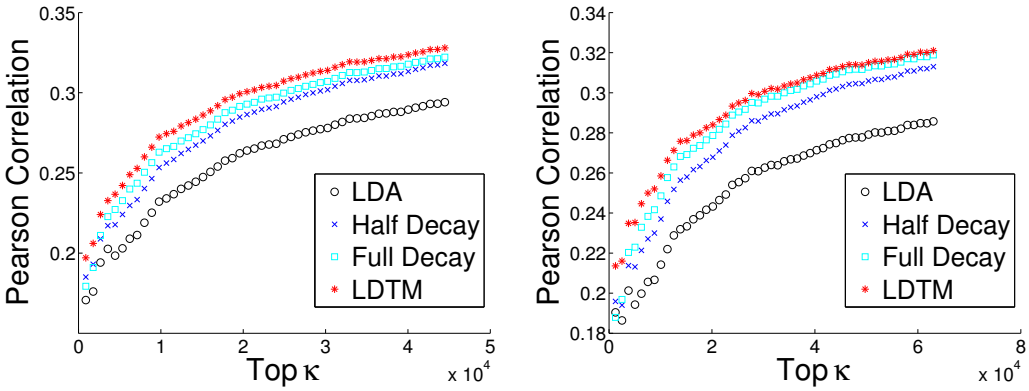
Figure 6.6: Creating Training Data Sets for Task 1 and Task 2

training data for Task 1 where some elements are missing as compared to the matrix in Figure 6.6(a). Using the training data, the objective of Task 1 is to predict all total adoptions for every user.

2. **Task 2:** For each user, we randomly select an item and hide the information in all time steps. Figure 6.6(c) shows the training data for Task 2 where the second and last item has been fully hidden. Using the training data, the goal of Task 2 is to predict the missing total adoptions for every user.

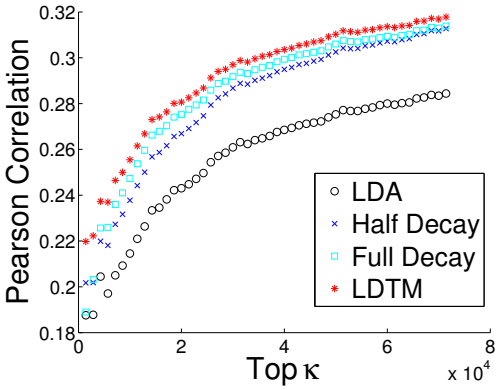
We repeated the prediction experiments for a total of five times and took the average results. In each run of the experiment, we generated five sets of

training and testing data by hiding in incremental proportions of 10%. from the sampled *ego-2* data sets. Each training set with a larger proportion of hidden data is derived from the previous training data. As a result, we obtained training sets with missing proportions of 10%, 19%, 27% and 34%.

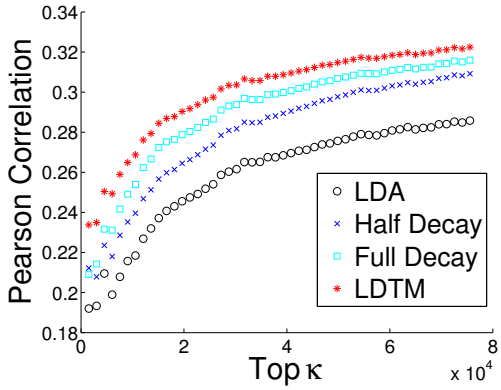


(a) Test size=10%

(b) Test size=19%



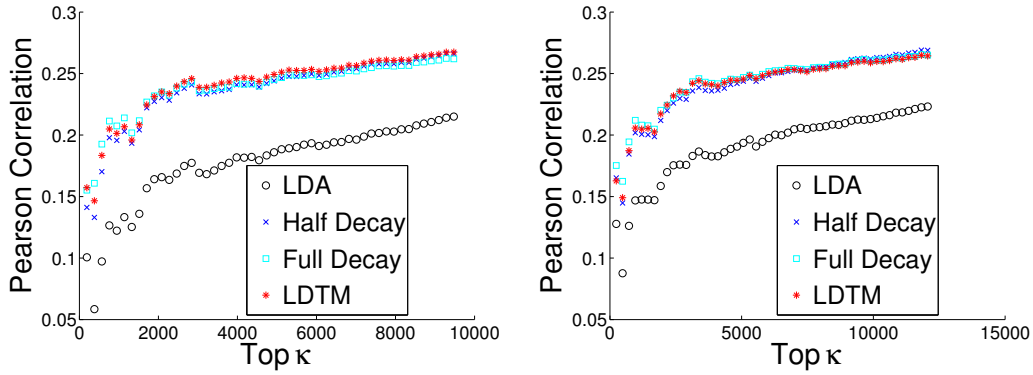
(c) Test size=27%



(d) Test size=34%

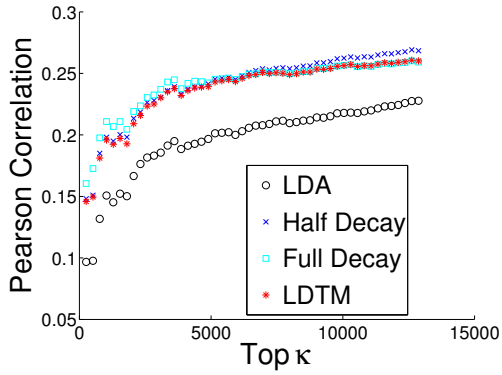
Figure 6.7: Pearson Correlation of Task 1 (ACMDL)

Figures 6.7 and 6.9 show the ACMDL results for Task 1 and Task 2 respectively. In both figures, the y-axis represents the Pearson Correlation Coefficient (PCC) between the actual frequency of user adopting item and the probability of user adopting item. We chose to compare PCC instead of Root Mean Squared Error (RMSE) because our model predicts the probability of adoption, not frequency. We compare the PCC for the top-κ test values and plot the results while varying κ using x-axis. The results show that LDTM outperforms all other baseline models with LDA performing the worst. This indicates that LDTM provides a good balance between information learned in

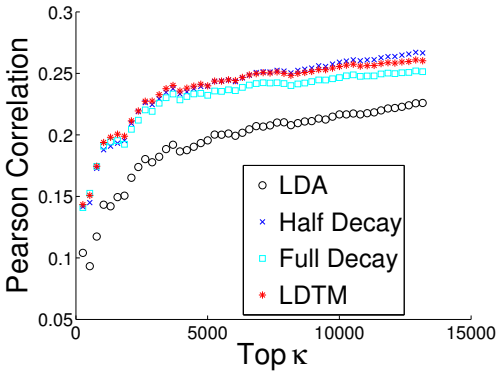


(a) Test size=10%

(b) Test size=19%



(c) Test size=27%



(d) Test size=34%

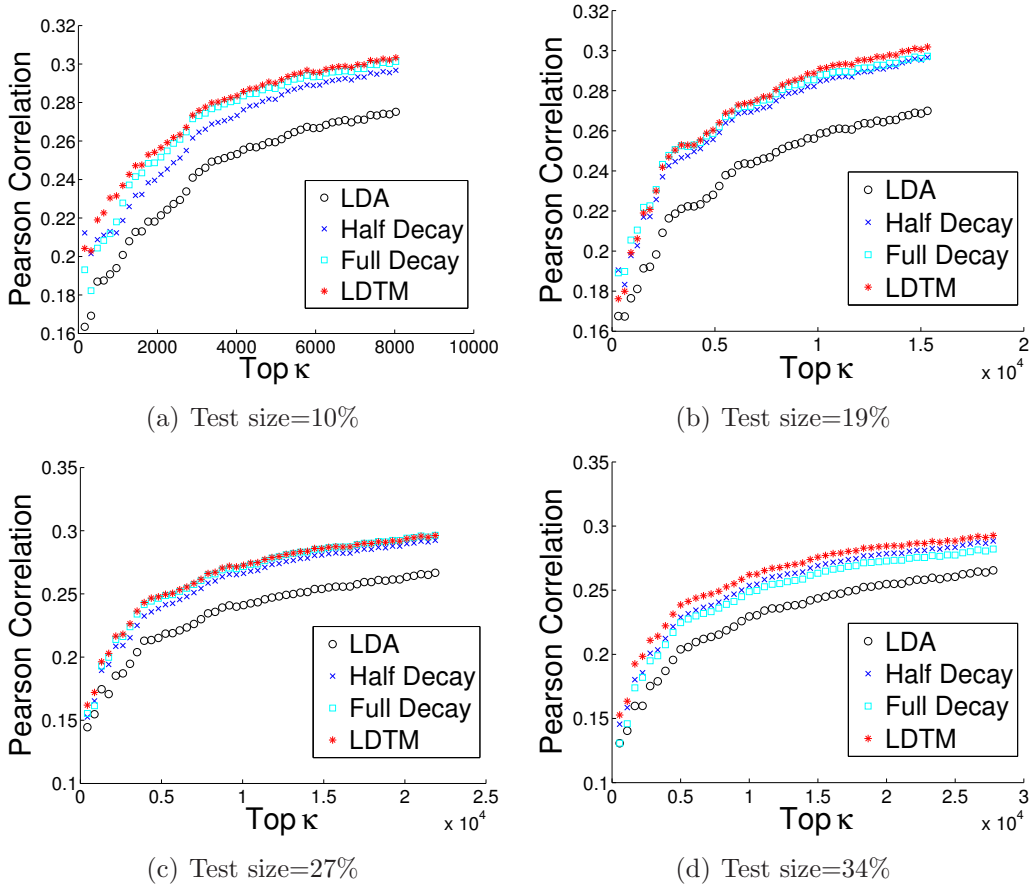
Figure 6.8: Pearson Correlation of Task 1 (DBLP)

past time steps and the present adoption information for obtaining a reliable estimation of adoption behavior for the user at each time steps.

We also observed that as κ becomes larger, the PCC increases more marginally. This suggests that we could only distinguish LDTM and the baseline models for large test values. In the case of DBLP, since we only have access to words in the title, adoption values for each time step of a user are significantly lower than ACMDL. As a result, the results of DBLP as shown in Figure 6.8 and 6.10 could not distinguish the performance of LDTM against other baseline models easily.

6.5.4 Results for TSC Evaluation

We evaluate our use of topic distributions as time series for computing the TSC between two authors i and j at a time point τ. The parameters "width" and "lookahead" are both set as 4. We apply our sampled *ego-3* data sets to



(a) Test size=10%

(b) Test size=19%

(c) Test size=27%

(d) Test size=34%

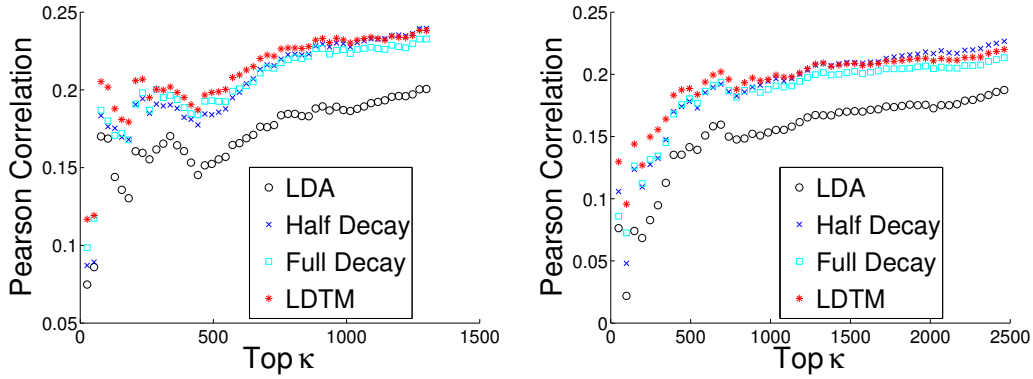
Figure 6.9: Pearson Correlation of Task 2 (ACMDL)

finding TSC by using the co-authorship information to choose pairs of authors and the year of publication as time point τ. For example, Figure 6.11 shows two authors interacted and wrote a paper together. We seek to find whether TSC is stronger from first author to the second author or vice versa.

We formulate the following hypotheses and perform Student's Paired T-test [98] for each hypothesis:

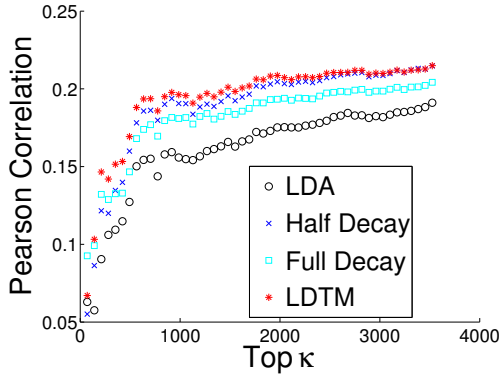
1. **AB**: If j is the first author and i is the second author of a publication written at τ, then i transfers information to j, i.e. $TSC(i \to j, \tau) > TSC(j \to i, \tau)$.

2. **AZ**: If j is the first author and i is the last author of a publication written at τ, then i transfers information to j, i.e. $TSC(i \to j, \tau) > TSC(j \to i, \tau)$.

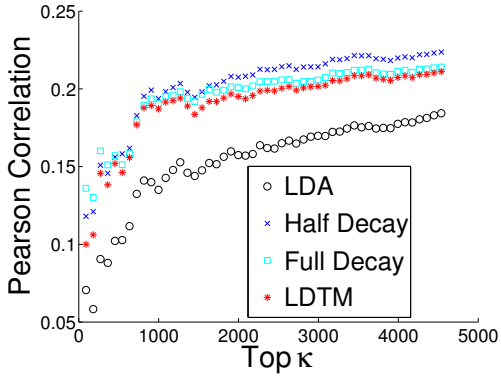


(a) Test size=10%

(b) Test size=19%



(c) Test size=27%



(d) Test size=34%

Figure 6.10: Pearson Correlation of Task 2 (DBLP)

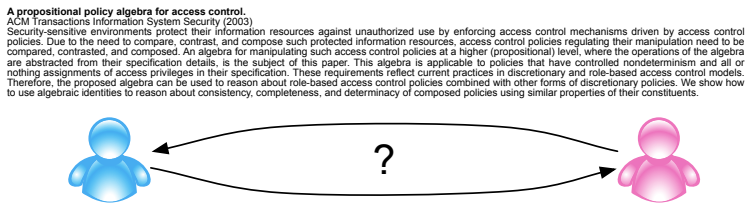


Figure 6.11: Example of Interaction Between Two Authors

3. **Bf_Af**: If j and i are authors of a publication written at τ with more than two authors and j comes before i, then i transfers information to j, i.e. $TSC(i \to j, \tau) > TSC(j \to i, \tau)$.

We compare for pairs of authors whose relationship in each hypothesis scenario exceeds more than four to ensure that author pairs have sustained interactions. For every i, j author pair who co-authored in multiple time steps, we compute the $TSC(i \to j, \tau), TSC(j \to i, \tau)$ for each time step τ where they co-authored a publication and take the averages of their TSC to obtain $TSC(i \to j)$ and

$TSC(j \rightarrow i)$. For every pair of authors, we arrange the computed TSC into two columns for each direction. We remove the bottom 2.5% and top 2.5% outliers from both columns. We then perform T-test [98] on both columns. T-test is used to determine whether both columns are significantly different from each other. In order to accept the three hypotheses proposed earlier, the T-test needs to show that both columns are significantly different and fulfill the condition we laid out earlier, in which TSC should be larger. In a T-test, smaller p-value indicates a more statistically significant result.

Table 6.2: T-tests of Hypotheses on Co-authors Relationship Using LDTM Topic Distribution

Hypo-thesis	Accept/ Reject	P-value	$TSC(i \rightarrow j) > TSC(j \rightarrow i)$	# of Pairs
ACMDL (ego-3)				
AB	Accept	9.91×10^{-10}	True	2,162
AZ	Accept	4.81×10^{-20}	True	2,326
Bf_Af	Accept	2.90×10^{-65}	True	13,580
DBLP (ego-3)				
AB	Accept	6.16×10^{-70}	True	20,568
AZ	Accept	6.15×10^{-121}	True	22,720
Bf_Af	Accept	5.04×10^{-253}	True	101,362

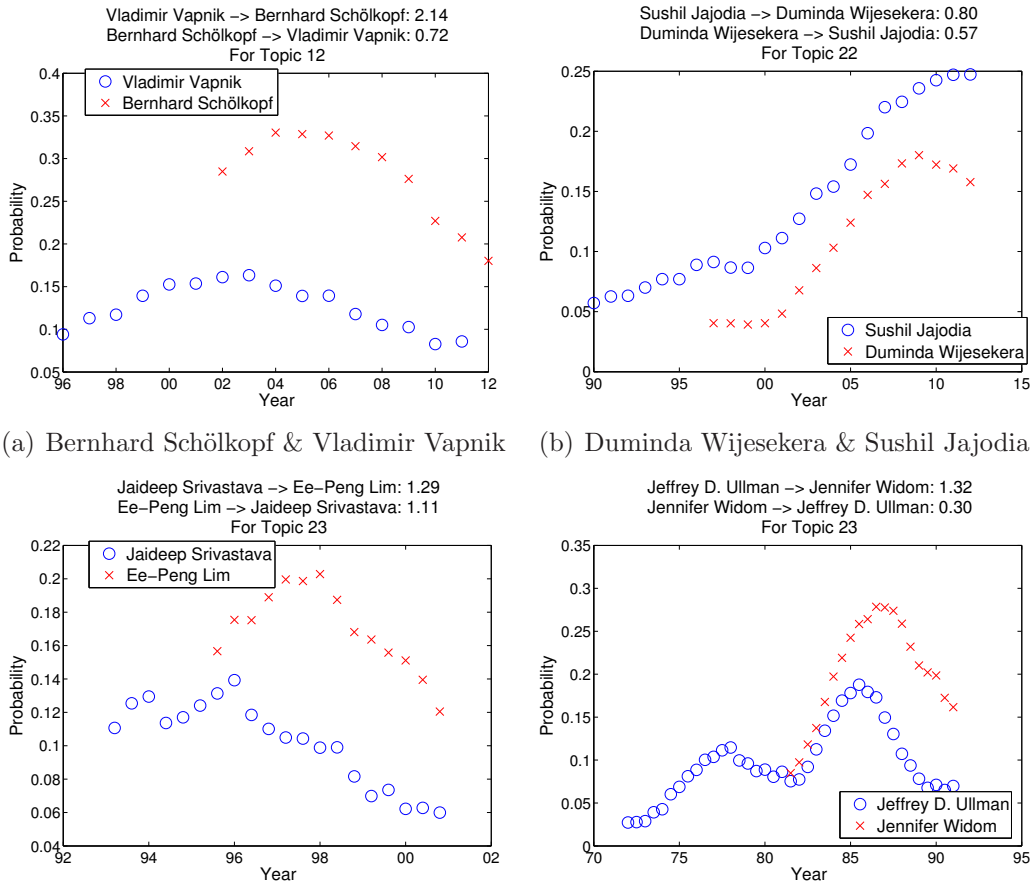
Table 6.2 shows the results of the hypothesis testing performed on all three hypotheses for ACMDL (ego-3) and DBLP (ego-3) data sets when using users' topic distributions from **LDTM**. For both data sets, we accept all three hypotheses which we defined earlier because the T-tests give a lower than 5×10^{-2} p-values and the respective columns in each of the hypothesis scenario is larger than the other column, i.e. $TSC(i \rightarrow j, \tau) > TSC(j \rightarrow i, \tau)$. Given that AB and AZ hypothesis scenarios have almost the same number of pairs but the p-value is significantly smaller in the AZ case. This suggests that the last author transfers more information to the first author as compared to the information the second author transfers to the first author. With the acceptance of Bf_Af, these hypotheses suggests that in academic publications, the i^{th} authors follows the j^{th} authors when $j > i$.

Table 6.3 shows the results of the hypothesis testing performed on all hypotheses for ACMDL (ego-3) and DBLP (ego-3) data sets when using users' topic distributions from **LDA**. However, we are not able to obtain a consistent acceptance or rejection of hypotheses when using users' topic distribution derived from LDA. There are two acceptance and four other rejections. Although T-test results provide low p-values, we reject four hypotheses because the TSC values did not meet our requirements. The p-values are also not as low as when we use the topic distributions derived from LDTM (cf. Table 6.2).

Table 6.3: T-tests of Hypotheses on Co-authors Relationship Using LDA Topic Distribution

Hypothesis	Accept/ Reject	P-value	$TSC(i \to j) > TSC(j \to i)$	# of Pairs
ACMDL (ego-3)				
AB	Reject	9.91×10^{-4}	False	2,150
AZ	Reject	3.59×10^{-14}	False	2,309
Bf_Af	Reject	1.87×10^{-13}	False	13,498
DBLP (ego-3)				
AB	Accept	3.55×10^{-2}	True	20,463
AZ	Reject	5.40×10^{-14}	False	22,625
Bf_Af	Accept	1.16×10^{-3}	True	100,913

While we do not have ground truth of evaluating the numerical accuracy of TSC values, T-tests on two different sets of topic distributions show that using LDTM topic distributions provide consistent outcomes for T-tests on both data sets and the different hypotheses scenarios. We counted the number of words written by authors at the time of their interaction τ. For a pair of authors (i, j) in the **AZ** scenario, the last author i had written **1.62** times more words than first author j. In the **AB** scenario, the second author i had written **1.42** times more words than j. This suggests that the last and second authors who had more adoptions prior to the point of interactions are the ones who transfer information to the first authors.



(a) Bernhard Schölkopf & Vladimir Vapnik (b) Duminda Wijesekera & Sushil Jajodia

(c) Ee-Peng Lim & Jaideep Srivastava (d) Jennifer Widom & Jeffrey Ullman

Figure 6.12: Case Studies

6.5.5 Case Studies

We highlight four examples of co-authors in Figure 6.12, who are prominent researchers in the areas of machine learning, privacy and security, data mining, and database.

- **Bernhard Schölkopf & Vladimir Vapnik** are renowned researchers in the field of Machine Learning. Vapnik is the inventor of the popular Support Vector Machine (SVM) algorithm while Schölkopf was Vapnik's PhD student. They co-authored 5 papers between the period of 1992 to 1997 with Schölkopf as first author and Vapnik as last author.

- **Duminda Wijesekera & Sushil Jajodia** are computer scientists and faculty members at George Mason University, Virginia, USA. Between 2001 to 2010, they have co-authored a total of 38 papers on computer

security and privacy.

- **Ee-Peng Lim & Jaideep Srivastava** are professors of computer science in database management, data mining and social networks. Lim was a former PhD student of Srivastava at the University of Minnesota, Minneapolis, USA, from 1989 to 1994. They have co-authored 11 papers from 1993 to 1998.

- **Jennifer Widom & Jeffrey Ullman** are Stanford computer science professors and leading researchers in database and data management. Widom worked closely with Ullman when she became a faculty member of Stanford University in 1993. They had published 19 papers together.

For each author pair, we chose the dominant topic from the first author and visualize the changes in topic distribution for both authors over a selected period of time. The dominant topic for an author is obtained by summing the topic distributions over the active time steps and choosing the topic with the highest value.

Table 6.4: Selected Topics with reference to Figure 6.12

Topic 12	Topic 22	Topic 23
learning	service	data
classification	management	database
recognition	security	mining
detection	scheme	processing
feature	internet	query

For purpose of visualization, the fluctuations in topic distributions are smoothen using eight year moving average. We choose eight due to the summation of four years width and four years lookahead. For the dominant topics in Figure 6.12, we also show the corresponding words of the topics in Table 6.4. In Figures 6.12(a) to 6.12(d), we may observe that by shifting the red "x" curves to the left, it merges with the blue "o" curves. This suggests that the trend of red "x" curves follows that of the blue "o" curves. The values at the

top of each Figure 6.12(a) to 6.12(d) also quantitatively show that the average $TSC(blue\text{“}o'' \rightarrow red\text{“}x'') > TSC(red\text{“}x'' \rightarrow blue\text{“}o'')$.

6.6 Summary

This work has contributed to the measurement of temporal social correlation based on actions (item adoptions) that users perform over time. We propose a linear dynamical topic model that synergizes the merits of probabilistic topic models and linear dynamical system in order to capture user adoption behavior over time. The EM algorithm for solving the model draws upon Gibbs Sampling, Kalman Filter, and RTS Smoothing for inference in the E-Step, followed by the M-Step which optimizes for the dynamics matrix. By taking into account both the stability and non-negativity constraints, we derive a dynamics matrix that represents how users decay their past preferences over time.

Furthermore, using the user topic distributions at different time steps, we construct each user's time series and compare it with their co-authors'. Employing Granger Causality on the time series, we then calculate the TSC between authors and discover that the ordering of authors' name on publication plays a role in how information transfers among the authors.

Chapter 7

Conclusion

7.1 Dissertation Summary

In the following paragraphs, we will take our readers through the journey of our accomplishments in this dissertation. When we started working on this topic, the existing Computer Science literature at that time had worked on finding social influence between social media users based on adoption of a single item [51, 52]. Instead of a single item, we wanted to infer social relationships between users based on the set of items the users adopt. We start off by analyzing static data since ignoring the time dimension greatly simplifies the analysis. The immediate questions we had were:

1. Is there any dependency between the social relationships users share and the common items they adopt?

2. How do we represent the behavior of each user based on the set of items each user adopts?

3. How do we model the social correlation using the behavior of the users and their social relationships?

4. How do we know whether the social correlation we found is accurate?

Chapter 3 is the outcome of our systemic inquiry on these questions. In Chapter 3, we answer the first question by showing the dependency using Contingency Tables and Fisher Exact Tests. We proposed the Sequential Social Correlation Model to answer the second and third question independently, then we proposed the Unified Social Correlation Model to answer both questions in an unified manner. To answer the fourth question, we use the user item adoption prediction. Due to the adoption prediction evaluation task, our work is often seen as part of the Collaborative Filtering literature.

The success of Chapter 3 led us to take a quantum leap by extending our static analysis to temporal data sets in Chapter 4. While we do not have to worry about the first question anymore, the remaining three questions continue to stumble our efforts in the temporal domain. We then proposed the Decay Topic Model (DTM) and the Two-period Temporal Social Correlation as an attempt to solve the social correlation problem on temporal data sets.

Two issues in Chapter 4 continue to pique our interests on the social correlation analysis for temporal data sets.

1. We made the assumption of having a constant decay parameter for Decay Topic Model (DTM) to simplify the temporal model. The model inference would be more elegant if we could let the dynamics of the data decide the decay parameter automatically.

2. The Two-period Temporal Social Correlation is a simplistic one because it only uses two time steps to infer the social correlation. This would result in social correlation values that would wildly fluctuate at different time steps for a pair of interacting users.

In Chapter 5, we looked into using Dynamic Matrix Factorization (DMF) to solve the constant decay parameter problem. Our intention was to use the Expectation Maximization (EM) algorithm [35] to optimize for the dynamic transition between the time steps of users' behavior. In the EM algorithm,

Kalman Filtering [49] and RTS smoothing [82] constitute the expectation (E-step) while the maximization (M-step) could optimize for the dynamics of transitions. But our first foray into the use of Matrix Factorization (MF) for representing users' behavior reveal an important insight that has not been discussed widely in the Computer Science literature. We discovered that non-negativity is necessary for a meaningful interpretation of the items' latent factors. The Gaussian distributions used in DMF and the EM algorithm does not allow us to obtain the users' latent factors and the non-negative item latent factors iteratively. While non-negative constraints such as log-barrier methods or Karush-Kuhn-Tucker (KKT) [50, 60] conditions could be added, the resulting equations would be far too complex to solve elegantly.

So in Chapter 6 we go back to using topic models for representing items' latent factors and users' latent factors which is similar to Chapter 4. But the experience of DMF in Chapter 5 now gives us the knowledge of how to automatically determine the decay parameter based on the dynamics of users' item adoption temporal data sets. We proposed Linear Dynamical Topic Model (LDTM) which is an aggregation of the accumulated knowledge from Chapters 3 through 5. We also wanted to go beyond the Two-period limitation on measurement of Temporal Social Correlation. We therefore utilized classical causality measures such as Granger causality which generalize the time window size. Granger causality takes our temporal users' behavior as inputs and calculate the F-statistic to obtain the Granger Causal Temporal Social Correlation. We then used the authors ordering in published academic papers as a proxy for the seniority of users to evaluate the perceived "influence".

The main contribution we see in this dissertation is that, while static dimension reductions and causality measures existed long before we started, the knowledge to bridge the two concepts together were absent in the literature, so this dissertation seeks to fill up the gap between the two.

7.2 Future Work

We conclude this dissertation by outlining several promising research directions for further improvements of the current work.

When we were conluding the work in Chapter 6, we realized that Linear Dynamical Topic Model could be simplified by omitting the use of RTS smoothing. The sole purpose of RTS smoothing is to allow data at later time steps to improve the inferences made in earlier time steps. RTS smoothing is relevant in the Control Theory/Engineering literature because they do not make changes to the observation matrix (the items' latent factor matrix in our case). But in our case, we are always updating the items' latent factor matrix (Topic item distribution) based on the new information in subsequent time steps. As a result of that, we only need the forward inference component (Kalman Filtering) while the backward inference is already taken care of by updating the items' latent factor.

We optimized for the decay parameters by making simplifying assumptions to reduce the disparity between the Euclidean distance of the parameters in the Dirichlet distributions between two time steps. The idea of minimizing the Euclidean distance between Dirichlet distribution parameters comes from the assumption that users do not drastically change their behaviors at different time steps. This assumption is to cope with temporal sparsity where users do not have adoptions at certain time steps. Instead of optimizing at the parameters of the Dirichlet distribution, we could minimize the Kullback-Leibler divergence between the Dirichlet distributions.

Ultimately, all the measurements we made is to further the science of predicting the future. However, it remains to be seen whether temporal social correlation could be used for making recommendations to users on what items to adopt. We could explore the use of social correlation for making predictions.

Bibliography

[1] Association for Computing Machinery. 2011. ACM Digital Library.

[2] Deepak Agarwal, Bee-Chung Chen, and Pradheep Elango. Fast online learning through offline initialization for time-sensitive recommendation. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 703–712, New York, NY, USA, 2010. ACM.

[3] Amr Ahmed, Yucheng Low, Mohamed Aly, Vanja Josifovski, and Alexander J. Smola. Scalable distributed inference of dynamic user interests for behavioral targeting. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 114–122, New York, NY, USA, 2011. ACM.

[4] 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.

[5] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. Influence and correlation in social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 7–15, New York, NY, USA, 2008. ACM.

[6] S. Aral and D. Walker. Identifying social influence in networks using randomized experiments. *Intelligent Systems, IEEE*, 26(5):91–96, 2011.

[7] Sinan Aral, Lev Muchnik, and Arun Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, 106(51):21544–21549, 2009.

[8] Eytan Bakshy, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 65–74, New York, NY, USA, 2011. ACM.

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

[10] Frank M. Bass. A new product growth for model consumer durables. *Manage. Sci.*, 50:1825–1832, December 2004.

[11] Chumki Basu, Haym Hirsh, and William Cohen. Recommendation as classification: using social and content-based information in recommendation. In *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, AAAI '98/IAAI '98, pages 714–720, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.

[12] Richard Ernest Bellman. *Dynamic Programming*. Dover Publications, Incorporated, 2003.

[13] J. C. Bezdek, R. J. Hathaway, R. E. Howard, C. A. Wilson, and M. P. Windham. Local convergence analysis of a grouped variable version of coordinate descent. *J. Optim. Theory Appl.*, 54(3):471–477, September 1987.

[14] Jeff Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report TR-97-021, ICSI, 1997.

[15] David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 113–120, 2006.

[16] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[17] Robert M. Bond, Christopher J. Fariss, Jason J. Jones, Adam D.I. Kramer, Cameron Marlow, Jaime E. Settle, and James H. Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295–298, 09/2012 2012.

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

[19] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and {ISDN} Systems*, 30(17):107 – 117, 1998. Proceedings of the Seventh International World Wide Web Conference.

[20] Alexander Brodsky, Csilla Farkas, and Sushil Jajodia. Secure databases: Constraints, inference channels, and monitoring disclosures. *IEEE Trans. on Knowl. and Data Eng.*, 12(6):900–919, November 2000.

[21] Kevin R. Canini, Lei Shi, and Thomas L. Griffiths. Online Inference of Topics with Latent Dirichlet Allocation. In *Proceedings of AI Stats*, 2009.

[22] Bin Cao, Dou Shen, Jian-Tao Sun, Xuanhui Wang, Qiang Yang, and Zheng Chen. Detect and track latent factors with online nonnegative matrix factorization. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 2689–2694, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[23] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna Gummadi. Measuring user influence in twitter: The million follower fallacy, 2010.

[24] 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.

[25] 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.

[26] Chen Cheng, Haiqin Yang, Irwin King, and Michael Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI*, 2012.

[27] Freddy Chong Tat Chua, Hady W. Lauw, and Ee-Peng Lim. Predicting item adoption using social correlation. In *SDM '11: Proceedings of the SIAM International Conference on Data Mining*. SIAM, 2011.

[28] Freddy Chong Tat Chua, Hady W. Lauw, and Ee-Peng Lim. Mining social dependencies in dynamic interaction networks. In *SDM*, Philadelphia, PA, USA, 2012. SIAM.

[29] Freddy Chong Tat Chua, Hady W. Lauw, and Ee-Peng Lim. Generative models for item adoptions using social correlation. *IEEE Trans. on Knowl. and Data Eng.*, 25(9):2036–2048, September 2013.

[30] Freddy Chong Tat Chua, Richard J. Oentaryo, and Ee-Peng Lim. Modeling temporal adoptions using dynamic matrix factorization. In *ICDM*, 2013.

[31] 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.

[32] 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.

[33] 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.

[34] Peng Cui, Fei Wang, Shiqiang Yang, and Lifeng Sun. Item-level social influence prediction with probabilistic hybrid factor matrix factorization. In *AAAI*, 2011.

[35] 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.

[36] Laura Dietz, Steffen Bickel, and Tobias Scheffer. Unsupervised prediction of citation influences. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 233–240, New York, NY, USA, 2007. ACM.

[37] 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.

[38] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Exploring temporal effects for location recommendation on location-based social networks. In *RecSys*, 2013.

[39] Sean Gerrish and David M. Blei. A language-based approach to measuring scholarly impact. In Johannes Frnkranz and Thorsten Joachims, editors, *ICML*, pages 375–382. Omnipress, 2010.

[40] Zoubin Ghahramani and Geoffrey E. Hinton. Parameter estimation for linear dynamical systems. Technical report, 1996.

[41] A. Gohr, A. Hinnerburg, R. Schult, and M. Spiliopoulou. Topic evolution in a stream of documents. In *SDM*, pages 859–870, Sparks, Nevada, 2009.

[42] Manuel Gomez Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 1019–1028, New York, NY, USA, 2010. ACM.

[43] 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.

[44] C. W. J. Granger. Essays in econometrics. chapter Investigating causal relations by econometric models and cross-spectral methods, pages 31–47. Harvard University Press, Cambridge, MA, USA, 2001.

[45] C.W.J. Granger. Testing for causality: A personal viewpoint. *Journal of Economic Dynamics and Control*, 2(0):329 – 352, 1980.

[46] S. Greenland. Randomization, statistics, and causal inference. *Epidemiology*, 1(6), 1990.

[47] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[48] M. D. Hoffman, D. M. Blei, and F. Bach. Online learning for latent dirichlet allocation. In *NIPS*, pages 1–9, 2010.

[49] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[50] William Karush. Minima of functions of several variables with inequalities as side conditions. Master's thesis, Department of Mathematics, University of Chicago, Chicago, IL, USA, 1939.

[51] 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.

[52] David Kempe, Jon Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *Proceedings of the 32nd international conference on Automata, Languages and Programming*, ICALP'05, pages 1127–1138, Berlin, Heidelberg, 2005. Springer-Verlag.

[53] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):pp. 81–93, 1938.

[54] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September 1999.

[55] 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.

[56] Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 447–456, New York, NY, USA, 2009. ACM.

[57] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[58] L. F. Kozachenko and N. N. Leonenko. Sample estimate of the entropy of a random vector. *Probl. Inf. Transm.*, 23(1-2):95–101, 1987.

[59] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Phys. Rev. E*, 69:066138, Jun 2004.

[60] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In Jerzy Neyman, editor, *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. University of California Press, Berkeley, CA, USA, 1950.

[61] 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.

[62] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[63] D. D. Lee and H. S. Seung. Algorithms for nonnegative matrix factorization. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 13, pages 556–562. MIT Press, 2001.

[64] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), May 2007.

[65] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 695–704, New York, NY, USA, 2008. ACM.

[66] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 631–640, New York, NY, USA, 2010. ACM.

[67] Michael Ley. DBLP Computer Science Bibliography, 2005.

[68] J. Lin. Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theor.*, 37(1):145–151, September 2006.

[69] Chao Liu, Hung-chih Yang, Jinliang Fan, Li-Wei He, and Yi-Min Wang. Distributed nonnegative matrix factorization for web-scale dyadic data analysis on mapreduce. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 681–690, New York, NY, USA, 2010. ACM.

[70] 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.

[71] Lu Liu, Jie Tang, Jiawei Han, and Shiqiang Yang. Learning influence from heterogeneous social networks. *Data Mining and Knowledge Discovery*, 25(3):511–544, 2012.

[72] Duc Luu, Ee-Peng Lim, Tuan-Anh Hoang, and Freddy Chua. Modeling diffusion in social networks using network properties, 2012.

[73] 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.

[74] 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.

[75] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, March 2010.

[76] Lev Muchnik, Sinan Aral, and Sean J. Taylor. Social influence bias: A randomized experiment. *Science*, 341(6146):647–651, 2013.

[77] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, Feb 2004.

[78] Juuso Parkkinen, Janne Sinkkonen, Adam Gyenge, and Samuel Kaski. A block model suitable for sparse graphs. In *Proceedings of the 7th International Workshop on Mining and Learning with Graphs (MLG 2009)*, Leuven, Belgium, July 2-4 2009. Extended Abstract.

[79] Judea Pearl. *Causality: models, reasoning and inference*, volume 29. Cambridge Univ Press, 2000.

[80] Huiming Qu, Jimeng Sun, and Hani T. Jamjoom. Scoop: Automated social recommendation in enterprise process management. In *Proceedings of the 2008 IEEE International Conference on Services Computing - Volume 1*, SCC '08, pages 101–108, Washington, DC, USA, 2008. IEEE Computer Society.

[81] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[82] H. E. Rauch, C. T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *Journal of the American Institute of Aeronautics and Astronautics*, 3(8):1445–1450, August 1965.

[83] Daniel M. Romero, Wojciech Galuba, Sitaram Asur, and Bernardo A. Huberman. Influence and passivity in social media. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6913 of *Lecture Notes in Computer Science*, pages 18–33. Springer Berlin Heidelberg, 2011.

[84] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.

[85] Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):pp. 41–55, 1983.

[86] Sam Roweis and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural Comput.*, 11(2):305–345, February 1999.

[87] D.B. Rubin. Bayesian Inference for Causal Effects: The Role of Randomization. *The Annals of Statistics*, 6(1):34–58, 1978.

[88] Ankan Saha and Vikas Sindhwani. Learning evolving and emerging topics in social media: a dynamic nmf approach with temporal regularization. In *Proceedings of the fifth ACM international conference on Web search*

and data mining, WSDM '12, pages 693–702, New York, NY, USA, 2012. ACM.

[89] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, Cambridge, MA, 2008. MIT Press.

[90] 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.

[91] W. R. Shadish, T. D. Cook, and Donald T. Campbell. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin, 2 edition, 2001.

[92] Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating word of mouth. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[93] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples (Springer Texts in Statistics)*. Springer, 3rd edition, May 2006.

[94] Sajid Siddiqi, Byron Boots, and Geoffrey Gordon. A constraint generation approach to learning stable linear dynamical systems. In *NIPS*, December 2007.

[95] Dan Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[96] 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.

[97] 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.

[98] STUDENT. The probable error of a mean. *Biometrika*, 6(1):1–25, 1908.

[99] J. Z. Sun, K. R. Varschney, and K. Subbian. Dynamic matrix factorization: A state space approach. In *IEEE International Conference on Speech and Signal Processing*, pages 1897–1900, 2012.

[100] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 807–816, New York, NY, USA, 2009. ACM.

[101] Jie Tang, Sen Wu, and Jimeng Sun. Confluence: conformity influence in large social networks. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '13, pages 347–355, New York, NY, USA, 2013. ACM.

[102] Martin Vejmelka and Milan Paluš. Inferring the directionality of coupling with conditional mutual information. *Phys. Rev. E*, 77:026214, Feb 2008.

[103] Greg Ver Steeg and Aram Galstyan. Information transfer in social media. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 509–518, New York, NY, USA, 2012. ACM.

[104] Greg Ver Steeg and Aram Galstyan. Information-theoretic measures of influence based on content dynamics. In *Proceedings of the sixth ACM*

international conference on Web search and data mining, WSDM '13, pages 3–12, New York, NY, USA, 2013. ACM.

[105] Jonathan D. Victor. Binless strategies for estimation of information from neural data. *Phys. Rev. E*, 66:051903, Nov 2002.

[106] Ken Wakita and Toshiyuki Tsurumi. Finding community structure in mega-scale social networks. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 1275–1276, New York, NY, USA, 2007. ACM.

[107] F. Wang, P. Li, and C. Konig. Efficient document clustering via online nonnegative matrix factorization. In *SDM*, pages 908–919, 2011.

[108] Lingyu Wang, Duminda Wijesekera, and Sushil Jajodia. Towards secure xml federations. In Ehud Gudes and Sujeet Shenoi, editors, *Research Directions in Data and Applications Security*, volume 128 of *IFIP The International Federation for Information Processing*, pages 117–131. Springer US, 2003.

[109] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 424–433, New York, NY, USA, 2006. ACM.

[110] 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.

[111] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 261–270, New York, NY, USA, 2010. ACM.

[112] Duminda Wijesekera and Sushil Jajodia. Policy algebras for access control: the propositional case. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, CCS '01, pages 38–47, New York, NY, USA, 2001. ACM.

[113] Duminda Wijesekera and Sushil Jajodia. Policy algebras for access control the predicate case. In *Proceedings of the 9th ACM conference on Computer and communications security*, CCS '02, pages 171–180, New York, NY, USA, 2002. ACM.

[114] Xin Xin, Irwin King, Hongbo Deng, and Michael R. Lyu. A social recommendation framework based on multi-scale continuous conditional random fields. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1247–1256, New York, NY, USA, 2009. ACM.

[115] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with Bayesian probabilistic tensor factorization. In *SDM*, pages 211–222, Columbus, OH, 2010.

[116] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 267–273, New York, NY, USA, 2003. ACM.

[117] 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.

[118] Jing Zhang, Biao Liu, Jie Tang, Ting Chen, and Juanzi Li. Social influence locality for modeling retweeting behaviors. In *Proceedings of the*

Twenty-Third international joint conference on Artificial Intelligence, IJCAI'13, pages 2761–2767. AAAI Press, 2013.

Appendix A

Additional Material for Static Social Correlation

A.1 Derivation of the E-Steps and M-Steps for Unified Generative Model

Suppose we have Θ the users latent factor distributions and Φ the latent factors item distribution. Then the likelihood of E is given by,

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

Then expressing in logarithm form,

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

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 Θ, Φ 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 $c_{u,x}$, $\theta_{x,z}$ and $\phi_{z,v}$:

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

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

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

The M Steps are,

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

A.2 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 A.1 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 A.2 shows a sample of the top

Table A.1: 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

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 A.2: 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

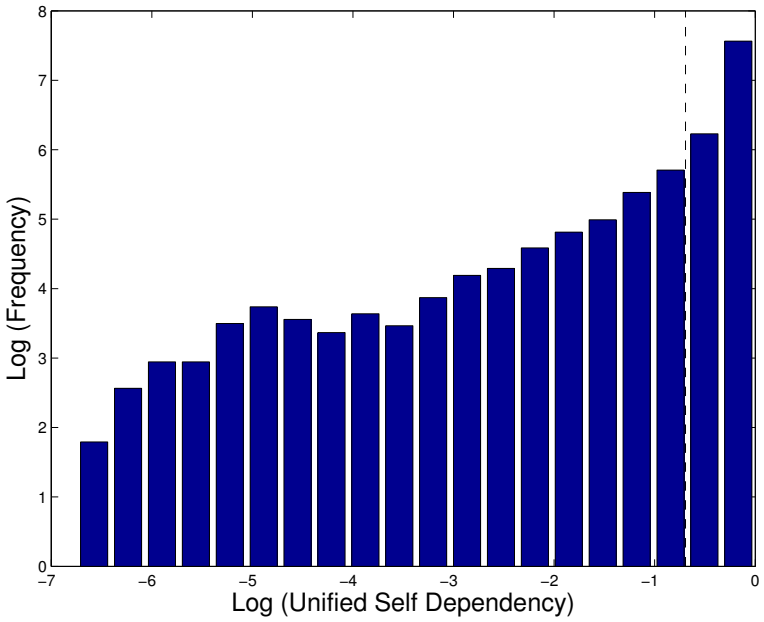


Figure A.1: LiveJournal: Histogram of Self Dependency

A.3 Distribution of Social Correlation

Figures A.1 and A.2 show the histogram of self-dependency values. The x-axis indicates the self-dependency values in logarithm scale and y-axis indicates 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.

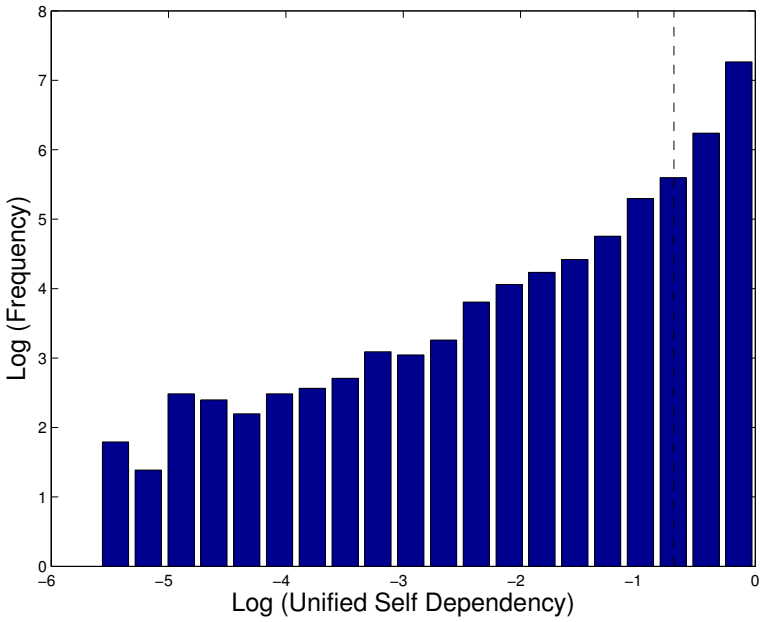


Figure A.2: Epinions: Histogram of Self Dependency

A.4 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,

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

$P(tp = t|k)$ 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.

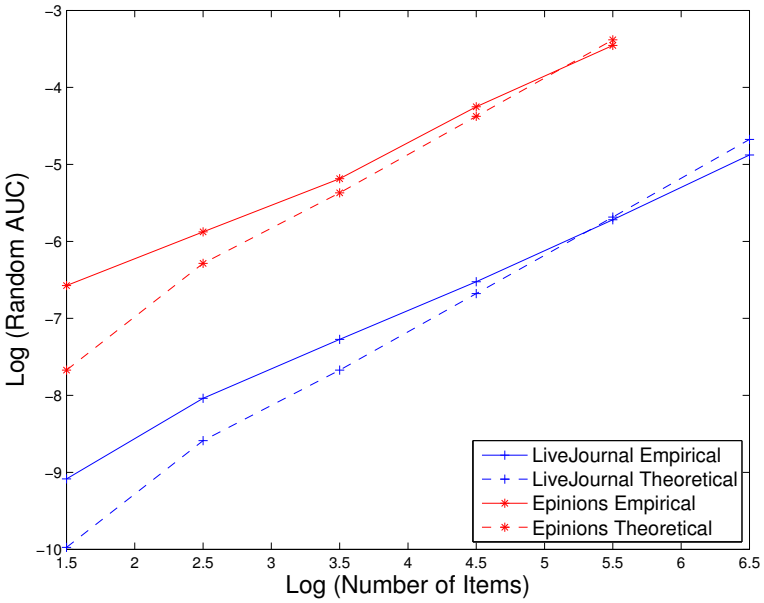


Figure A.3: Log(AUC of Random) vs Log(Number of Items)

Figure A.3 shows the theoretical and actual empirical results given by *Ran-*

dom. 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 A.3 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 3.4 and 3.6 are relatively higher than *Random*.

Appendix B

Additional Material for Linear Dynamical Topic Model

B.1 Derivation of the Smoothed Parameters

To obtain the necessary quantities using RTS smoothing,

$$x_{n,t|T} = x_{n,t|t} + J_{n,t}\left(x_{n,t+1|T} - x_{n,t+1|t}\right)$$

$$J_{n,t} = V_{n,t|t}A'_{n,t}V^{-1}_{n,t+1|t}$$

$$V_{n,t|T} = V_{n,t|t} + J_{n,t}\left(V_{n,t+1|T} - V_{n,t+1|t}\right)J'_{n,t}$$

$$V_{n,t+1,t|T} = V_{n,t+1|t+1}J'_{n,t} + J'_{n,t+1}\left(V_{n,t+2,t+1|T} - A_{n,t+1}V_{n,t+1|t+1}\right)J'_{n,t}$$

The expression for $V_{n,t+1,t|T}$ is a recursive equation that depends on $V_{n,T,T-1|T}$. Details of obtaining $V_{n,T,T-1|T}$ is given in Appendix B.2. The other quantities needed for RTS smoothing is given by Kalman Filtering,

$$x_{n,t|t-1} = A_{n,t-1}x_{n,t-1|t-1}$$

$$V_{n,t|t-1} = A_{n,t-1}V_{n,t-1|t-1}A'_{n,t-1} + Q$$

$$x_{n,t|t} = x_{n,t|t-1} + \psi_{n,t}$$

Here the missing quantity is $V_{n,t|t}$ and its derivation is,

$$\begin{aligned}
V_{n,t|t} &= E\left[\left(x_{n,t} - x_{n,t|t}\right)\left(x_{n,t} - x_{n,t|t}\right)'\right] \\
&= E\left[\left(x_{n,t} - x_{n,t|t-1} - \psi_{n,t}\right)\left(x_{n,t} - x_{n,t|t-1} - \psi_{n,t}\right)'\right] \\
&= E\left[\left(x_{n,t} - x_{n,t|t-1}\right)\left(x_{n,t} - x_{n,t|t-1}\right)' - 2\left(x_{n,t} - x_{n,t|t-1}\right)\psi_{n,t}' + \psi_{n,t}\psi_{n,t}'\right] \\
&= V_{n,t|t-1} + \psi_{n,t}\psi_{n,t}'
\end{aligned}$$

B.2 Initial Value of the Lag-One Covariance Smoother

Using the following relationship,

$$x_{n,t|T} = x_{n,t|t} + J_{n,t}\left(x_{n,t+1|T} - x_{n,t+1|t}\right)$$

$$x_{n,t|T} - x_{n,t} = x_{n,t|t} - x_{n,t} + J_{n,t}\left(x_{n,t+1|T} - x_{n,t+1|t}\right)$$

To obtain $V_{n,t+1,t|T}$, we define the following,

$$V_{n,t+1,t|T} = E\left[\left(x_{n,t+1} - x_{n,t+1|T}\right)\left(x_{n,t} - x_{n,t|T}\right)'\right]$$

The initial value, $V_{n,T,T-1|T}$ is given by,

$$V_{n,T,T-1|T} = E\left[\left(x_{n,T} - x_{n,T|T}\right)\left(x_{n,T-1} - x_{n,T-1|T}\right)'\right]$$

$$x_{n,T|T} = x_{n,T|T-1} + \psi_{n,t}$$

$$x_{n,T-1|T} = x_{n,T-1|T-1} + J_{n,T-1}\left(x_{n,T|T} - x_{n,T|T-1}\right)$$

$$V_{n,T,T-1|T} = E\Big\{\left[(x_{n,T} - x_{n,T|T-1}) - \psi_{n,t}\right]$$

$$\left[(x_{n,T-1} - x_{n,T-1|T-1}) - J_{n,T-1}\left(x_{n,T|T} - x_{n,T|T-1}\right)\right]'\Big\}$$

$$= E\left\{\left[(x_{n,T} - x_{n,T|T-1}) - \psi_{n,t}\right]\left[(x_{n,T-1} - x_{n,T-1|T-1}) - J_{n,T-1}\psi_{n,t}\right]'\right\}$$

$$= E\left[(x_{n,T} - x_{n,T|T-1})(x_{n,T-1} - x_{n,T-1|T-1})' + \psi_{n,t}\psi_{n,t}'J_{n,T-1}'\right]$$

$$= V_{n,T,T-1|T-1} + \psi_{n,t}\psi_{n,t}'J_{n,T-1}'$$

We prove the following,

$$V_{t+1,t|t} = AV_{t|t}$$

The right-hand side can be evaluated as,

$$AE\left[\left(x_t - x_{t|t}\right)\left(x_t - x_{t|t}\right)'\right]$$

$$= E\left[\left(Ax_t - Ax_{t|t}\right)\left(x_t - x_{t|t}\right)'\right]$$

$$= E\left[\left(x_{t+1} - x_{t+1|t}\right)\left(x_t - x_{t|t}\right)'\right]$$

$$= V_{t+1,t|t}$$

By showing the right-hand side equals to the left-hand side, we finally obtain the initial value,

$$V_{T,T-1|T} = AV_{T-1|T-1} + \psi_t\psi_t'J_{T-1}'$$