2-2017

Modeling adoption dynamics in social networks

Minh Duc LUU
Singapore Management University, mdluu.2011@phdis.smu.edu.sg

Citation

LUU, Minh Duc. Modeling adoption dynamics in social networks. (2017). Dissertations and Theses Collection.
Available at: http://ink.library.smu.edu.sg/etd_coll_all/4

MODELING ADOPTION DYNAMICS IN SOCIAL NETWORKS

LUU MINH DUC

SINGAPORE MANAGEMENT UNIVERSITY

2016

Modeling Adoption Dynamics in Social Networks

by

Luu Minh Duc

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

Dissertation Committee:

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

David Lo
Assistant Professor of Information Systems
Singapore Management University

Hady Wirawan Lauw
Assistant Professor of Information Systems
Singapore Management University

Qirong Ho
Scientist I
Institute for Infocomm Research

Singapore Management University
2016

Modeling Adoption Dynamics in Social Networks

by

Luu Minh Duc

Abstract

This dissertation studies the modeling of user-item adoption dynamics where an item can be an innovation, a piece of contagious information or a product. By "adoption dynamics" we refer to the process of users making decision choices to adopt items based on a variety of user and item factors. In the context of social networks, "adoption dynamics" is closely related to "item diffusion". When a user in a social network adopts an item, she may influence her network neighbors to adopt the item. Those neighbors of her who adopt the item then continue to trigger more adoptions. As this progress unfolds over time, the item is diffused through the social network. This connection motivates us to study also item diffusion modeling.

The factors which can affect user-item adoption include (i) *user* factors, e.g., interests, budget constraints, and brand preference; (ii) *item* factors, e.g., item features and brands; (iii) *social* factors, i.e., social influence from friends adopting and/or making recommendations on certain items; and (iv) external factors. The external factors are all other factors which we cannot observe and infer from data. Examples include marketing campaigns and advertisements. This thesis therefore focuses on user, item and social factors only.

Modeling how the three kinds of factors interact with one another as item adoptions occur is an important research problem as these factors can be utilized in search and recommendation applications. Ideally, we would like to incorporate all

these factors in a single model but the resultant model will be highly complex and computationally expensive. Thus, we first model adoption dynamics without social factors. Moreover, we focus on those user and item factors that are related to brands. Brands not only shape consumer perceptions of item quality but also the status associated with the ownership of item [1, 36, 37]. Product branding and brand management are also important marketing research topics but have long been overlooked in item adoption modeling and prediction research. In the first part of this work, we thus identify two novel brand-related factors, namely (i) *brand consciousness* of users, and (ii) *exclusiveness* of item brands. The former is a user factor and the latter is an item factor. We incorporate them into a topic model for item adoptions. The resultant model, called Brand-Item-Topic (BIT), not only improves remarkably adoption prediction accuracy but also returns actionable insights about users and items. We later develop a distributed and enhanced version of BIT, called DeBIT, which further achieves a linear scale-up and improves prediction accuracy.

In the second part of this work, we add social factor to the modeling of user-item adoptions by creating a matrix factorization model, called Social Brand-Item-Topic (SocBIT), which jointly models brand and social effects. Experiments on real data show that SocBIT improves the adoption prediction accuracy over the state-of-the-art social recommendation models such as SoRec [85] and RSTE [86].

In real life, multiple items are diffused together in the same social network. Diffusion of one item may boost or impede that of another, depending on how similar or dissimilar they are. Nevertheless, existing models of diffusion have been built upon *independent* contagion assumption whereby diffusion of each item is assumed to occur independently from other items. Thus, in the third part of this dissertation, we propose a novel framework for multi-item diffusion considering item interaction. The framework also incorporates homophily, the tendency that users connect to similar others. Based on the framework, we then propose a model, called Topic-level Interaction Homophily Aware Diffusion, which performs very well in Twitter hashtag diffusion prediction task.

Contents

List of Figures

List of Tables

Acknowledgement

The completion of this dissertation is made possbile by the assistance given to me by several persons whom I am very much grateful to. It is hard to quantify the amount of support I receive from them. The least I can do is to acknowledge them in this short little section and to do my best to contribute back to research community and society.

First of all, I feel fortunate to have a stable environment in SMU to pursue my PhD. I am indebt to Singapore providing me the scholarship support throughout my study. This allows me to concentrate fully on my dissertation research.

I was introduced to Professor Ee-Peng Lim by Professor Hady Wirawan Lauw in early 2011 and started my Ph.D. in the same year under the supervision of Professor Lim. Professor Lim not only provides me valuable advice and guidance but also gives me strong mental support during my long Ph.D. journey. I am thankful to Living Analytics Research Centre (LARC) for the exchange program at Carnegie Mellon University (CMU), where I had a collaboration with Professor Andrew C. Thomas which resulted in a jointly authored paper.

During my candidature, I have had the opportunity of collaborating with several brilliant researchers. Freddy C. T. Chua gave me useful advice in my early work on diffusion modeling. The collaboration led to two research papers. My later collaboration with Dr. Qirong Ho allowed me to develop efficient topic models using distributed computation.

I would like to thank the following individuals in LARC for both friendships and research guidance: Richard J. Oentaryo, Palakorn Achananuparp, Dai Bing

Tian, Philips K. Prasetyo, Agus T. Kwee, Arinto Murdopo, Ibrahim Nelman Lubis, David Low, Freddy C. T. Chua, Tuan-Anh Hoang and Gong Wei.

I also would like to thank the staff members of LARC and SIS for their excellent administrative support: Fong Soon Keat, Phoebe Yeo, Desmond Yap, Nancy Beatty; Ong Chew Hong, Seow Pei Huan and many others whom I have no opportunity to know them better.

I would like to thank the Singapore National Research Foundation for the PhD support. The research leading to the completion of this dissertation is funded by The Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

Finally, I must also thank my family members especially my wife and mother for the patience and support throughout my PhD. Without them, I would not have been able to focus on the PhD projects.

Publications

Publications based on the disssertation. Listed by reverse chronological order:

1. Minh-Duc LUU and Ee-Peng LIM, *Do Your Friends Make You Buy This Brand? Modeling Social Recommendation with Topics and Brands*, (under review in Data Mining and Knowledge Discovery, DMKD'16). (Chapter 4)

2. Minh-Duc LUU and Ee-Peng LIM, *Latent Factors Meet Homophily in Diffusion Modeling*, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery, ECML-PKDD'15. (Chapter 5).

3. Minh-Duc LUU, Ee-Peng LIM and Freddy Chong Tat CHUA, *On Modeling Brand Preferences in Item Adoptions*, AAAI International Conference on Web and Social Media, ICWSM'14. (Chapter 3).

Other publications during Ph.D. study.

1. Minh-Duc LUU and Andrew C. THOMAS, *Beyond mere following: Mention network, a better alternative for researching user interaction and behavior*, Social Computing, Behavioral-Cultural Modeling and Prediction, SBP'15.

2. Minh-Duc LUU, Ee-Peng LIM, Tuan-Anh Hoang and Freddy Chong Tat CHUA, *Modeling Diffusion in Social Networks Using Network Properties*, ICWSM'12.

Chapter 1

Introduction

1.1 Motivation

Nowadays, unprecedented growth of social media, online retail and other online services has made it possible for users to perform all kinds of online item adoptions including product purchases, movie downloads, restaurant visits, university applications, to name a few. With users connected with one another, the adoption of an item by a user may actually cause her social friends to adopt the same item. As a result, the item adoption behavior diffuses across the network. In this thesis, we therefore study two item adoption dynamics and they are: (a) non-diffusing item adoption; and (b) diffusing item adoption.

The decision processes behind item adoption dynamics are usually associated with (i) **user factors**, e.g., user topic interest, brand preference and attributes such as age, gender, etc., and (ii) **item factors**, e.g., item price, item brand and item popularity. As shown in the following works, these factors are very important to not only improving accuracy of adoption prediction but also to the understanding of user decision process. Such knowledge facilitates various applications such as user segmentation [122, 130], personalized recommendation [104] and targeted marketing [42, 122].

- Several successful recommendation models rely mainly on learning such user and item factors. For instance, the Netflix prize winning model (Koren *et al.* [67, 68]) factorized the movie rating matrix to infer (i) movie factors e.g. comedy versus drama, amount of action, or orientation to children, and (ii) user factors such as genre preference, amount of action he/she requires and so on.

- The works [1, 36, 37] by Aaker *et al.* show the importance of brands in terms of providing a quality guarantee, shaping a user's perception and associating social status to item's owner. This applies to both online and offline retail [116].

However, many of these factors are usually unobservable from adoption data. Thus, it is necessary to develop models for *adoption dynamics*, a.k.a. the process of users making decisions to adopt items, to infer such latent factors from observed adoption data and to utilize these factors for future adoption prediction and recommendation.

Matrix Factorization (MF) and Topic Modeling (TM) are two well established approaches for inferring such latent factors from non-diffusing adoption data. Both approaches rely on the intuitive assumption that users tend to adopt and/or give high ratings to items which are *similar* to them. By similarity between an item and a user, we mean that the item possesses factors which are similar to the user's preference. MF represents adoption data as a user-item matrix and seeks to find a factorization of the matrix into two low-rank matrices which represent user and item latent factors respectively. The representation allows to estimate user-item similarity, e.g. by cosine similarity between their vector representations. Meanwhile, topic models associate each user with a distribution of latent topics indicating her topic preferences. By imposing non-negative constraints, the Non-negative Matrix Factorization model [71] can infer latent factors which are similar to latent topics in topic models. Thus, both approaches can be adapted to model item adoption by assuming that each user adopts items matching her topic preferences.

However, such approaches overlook one important aspect: *brand effects* in adoption. Brands can affect adoption behavior as they not only shape consumer perceptions of item quality but also the status associated with the ownership of item (see [1, 36, 37, 116]). Learning such brand effects provides fine-grained actionable insights on users and the brands themselves. For instance, there are users who are strongly affected by brand names and thus enjoy adopting items from brand names. It is important for vendors/suppliers to discover such users as they are usually willing to pay more for branded items. Meanwhile, there are brands who are *exclusively* adopted by a group of "elite" users — e.g. luxury fashion brands and cars, elite universities (as academic "brands"). Finding such exclusive brands, or more generally estimating brand exclusiveness, is very beneficial for applications such as brand ranking, brand marketing and product recommendation. Inspired by such brand importance, in this dissertation, we first incorporate brand-related factors into user-item adoption modeling to improve adoption prediction accuracy and to obtain valuable insights on adoption decision process of users. The brand related factors are **brand consciousness** (user factor) and **brand exclusiveness** (item factor).

In social networks, especially online social networks (OSNs), social influence from a user's neighbors usually has remarkable effect on how the user evaluates and/or adopts items. For example, we usually consult our family members, friends or acquaintances for item recommendations. We refer to these as **social factors**. Thus, our next step is to incorporate such social factors into modeling adoption dynamics. Inspired by the previous argument on the importance of brand effect, we aim to develop a joint model which unifies both brand-related user, item factors and social factors.

In the context of OSNs, user-item adoption dynamics includes item diffusion. When a user in an OSN adopts an item, she may influence her neighbors in the network to adopt the item. Those neighbors of her who actually adopt the item then continue to trigger more adoptions. This progress unfolds over time and the item is diffused through the OSN. In other words, diffusion can be considered as a process of social influence among adopters and non-adopters over time. Modeling item diffusion is thus necessary to get a better understanding on the impact of social influence on adoption dynamics. Such knowledge is then useful for applications such as Viral Marketing, Influence Maximization/Minimization [18, 62, 65].

In real life, diffusion of one item may boost or impede that of another. For instance, while the diffusion of iPhones in the Facebook friendship network may boost that of iPad, it may impede the diffusion of Android phones. However, most existing models of diffusion are built upon *independent* contagion assumption whereby the diffusion of each item is assumed (at least implicitly) to happen independently from other items. The interaction among items during diffusion is thus left out of the picture. Modeling these interactions is crucial in both theory and practice since it helps us understand the detailed dynamics of multi-item diffusion. It is also valuable for business to develop suitable strategies to promote diffusion of their own items considering the other items that have been diffused recently or are being diffused. Another important aspect which also has great impact on item diffusion, is the well-known *homophily* phenomenon, which refers to the tendency of individuals to associate with similar others. It is well known that homophily affects the mechanisms in which item diffusion happens, be it innovation [108], information [27] or behavior [20]. Thus, it is important to integrate homophily into diffusion models so that we can better quantify its effect on diffusion. We need a modeling framework for *multi-item* diffusion under effects of *item interaction* and *homophily*. The framework is expected (i) to fill the gap in existing diffusion literature with very few models considering such effects, and (ii) to be effective in terms of improving accuracy of adoption prediction.

1.2 Research Objectives

In short, the dissertation focuses on the following research objectives, namely, (i) incorporating brand effects into modeling adoption dynamics to learn the brand related latent factors that affect user adoption behavior, and (ii) modeling multi-item diffusion in a social network under the effects of item interaction and homophily. We provide a high level description of our approaches toward achieving the objectives and the challenges to be addressed in the following sections.

1.2.1 Incorporating Brand Effects into Adoption Modeling

We first define the concept of *brand* to distinguish it from other attributes of items. This definition will be used throughout this dissertation. Specifically, we adapt the following definition of *brand* from American Marketing Association (AMA) dictionary[1]: “A brand is a name, term, design, symbol, or any other feature that identifies one seller’s good or service as distinct from those of other sellers.” This definition can be generalized as following.

Definition 1. *A brand is an entity which creates items/services that can be differentiated from the items/services of others.*

Under this definition, not only companies can be considered as brands, but individuals are also qualified as “brands”. For instance, Steven Spielberg is a “brand” among movie directors as he creates blockbuster movies, Jackie Chan is a “brand” of fun kung-fu movies.

To incorporate the brand effects, we first extend topic models (TM) [13, 51]. TM associates (i) each user with a topic distribution indicating her topic interest, and (ii) each topic with an item distribution indicating the inherent content of the topic as well as popularity scores of items under the topic. Thus TM only captures *topic-based* user-item adoption, henceforth we extend it by introducing *brand-based* user-item adoption, whereby a user first selects a topic of her interest, then selects a brand under the topic and finally adopts some item under the brand. Compared with standard topic models such as Latent Dirichlet Allocation (LDA) [13], this formulation creates additional distributions such as topic-brand and brand-item distributions. Model inference thus gets much more complicated than LDA. We tackle this

[1] https://www.ama.org/resources/Pages/Dictionary.aspx?dLetter=B

technical challenge by developing an inference algorithm based on Gibbs sampling. Although the algorithm is demonstrated to work well on moderate-size datasets, it is not scalable to large datasets. We thus follow the so-called Stale Synchronous Parallel framework [49] to develop a scalable version of the algorithm.

In the context of online social networks, modeling adoption dynamics without considering social factors is obviously inadequate. We thus aim to jointly model both brand effect and social factors in user-item adoption. To achieve this goal, we can integrate the relevant user, item and social factors into some topic model or MF model. In this dissertation, we take the latter by approach by incorporating brand-related factors into well known *social recommendation* MF based models [85, 86, 91], which already possess the capability of handling social factors in user-item adoption dynamics. The resultant model from this approach has a simpler inference procedure which can be formulated as a Maximum A Posteriori (MAP) problem and solved by gradient descent method.

1.2.2 Modeling Multi-item Diffusion Considering Item Interaction and Homophily

To model multi-item diffusion with item interaction, we combine the essence of matrix factorization (MF) models and threshold models [45, 112], a family of well-established single-item diffusion models. Specifically, we incorporate into MF models the following variant of social influence component from threshold models. Originally, threshold models assume that each user can be triggered to adopt an item if the amount of social influence from her neighbors who adopted *the same item* exceeds a certain threshold. In our framework, the social influence comes from not only neighbors who adopted the same item but also neighbors who adopted *similar items* recently. Thus, recent adoptions of items similar to a given item can increase the probability that the item itself is now adopted. By this way, our framework can capture the fact that diffusion of items similar to a given item can boost the diffusion of the item itself.

Finally, to incorporate homophily, we explicitly model social influence exerted by a neighbor v on a user u as an increasing function of the similarity between u and v. Moreover, the function is parameterized by a global value indicating the *homophily level* of the whole network. The higher the homophily level, the stronger the connection between similar friends.

1.3 Contributions

To summarize, we list major contributions of this dissertation in the following.

1. To model and predict adoptions more accurately as well as obtain more insights on adoption decision process, we propose a new model for item adoption which incorporates brand-related factors such as *brand exclusiveness* and *brand-consciousness*. The former involves item factor and the latter involves user factor. To the best of our knowledge, our model, **Brand-Item-Topic** (**BIT**), is the first to consider these factors in modeling adoption. This piece of work is described in Chapter 3.

2. We develop a rigorous inference algorithm for BIT using Gibbs sampling and demonstrate that inferred model parameters not only provide actionable insights on user and item factors but also can improve significantly accuracy of item adoption prediction. Moreover, we also develop a **D**istributed and **e**nhanced version of BIT, called **DeBIT**. The new version is demonstrated to be scalable to real-world datasets and improves even further both the accuracies of parameter learning and of prediction. We also compare the user brand-consciousness and brand exclusiveness inferred by BIT and DeBIT against brand-related empirical measures to show the validity of these measures. We cover the proposed DeBIT and its comparison with BIT in Chapter 3.

3. Inspired by the success of BIT and DeBIT, we propose in Chapter 4 a joint model which incorporates both brand and social effects into adoption and/or

rating modeling. The model, called **SocBIT**, is shown to improve significantly the accuracy of adoption prediction over state-of-the-art models SoRec [85] and STE [86]. The improvement is at least 30% on restaurant adoption data (from Foursquare) and 20% on paper adoption data (from ACM Digital Library).

4. We propose in Chapter 5 a general framework for multi-item diffusion considering item interaction and homophily. Based on the diffusion framework, we derive a specific model, called *Topic level Interaction Homophily Aware Diffusion* (TIHAD). The model is able to capture diffusion processes of an *arbitrary* number of interacting items and also yields higher accuracy than baselines in an experiment on Twitter hashtag adoption prediction.

Chapter 2

Related Work

In this chapter, we first provide an overview on brand effect in user-item adoption, which has actually motivated us to incorporate brand-related factors into adoption modeling. We then provide a taxonomy of models for adoption dynamics. The taxonomy plays the role of a roadmap and some of it models which are most closely related to ours will be reviewed in details in subsequent sections. We move on to review two well established families of models for inferring user and item latent factors from observed adoption data, namely *Topic Models* and *Matrix Factorization* models. The two families of models provide the necessary basic knowledge for our proposed models in Chapters 3 and 4. Finally, we review existing diffusion models which consider item interaction and relate those models with our proposed model in Chapter 5.

2.1 Brand Effect in Adoption Dynamics

In traditional offline markets, the importance of brands has been recognized in marketing literature in numerous studies [1, 2, 17, 31, 35, 36, 37, 38, 53, 58, 97, 123]. Brands play a major role in adoption process since they help to shape consumer perceptions and tastes, thus inspire adoption behavior. In this way, strong brands not only create demand but also continuity of demand into the future by leveraging consumer's favorable preference to the brands. Moreover, authors in [1, 2, 31] noted that brand preference leads to other marketing advantages such as favorable word of mouth, reduced marketing cost. These studies confirm the importance of determining strong brands and users who are attracted to brands, thus motivate our research.

In online markets, brands continue to show their importance [35], [37], [116]. By comparing the effects of pricing in online shopbots (i.e. internet services that compare prices of similar consumer goods sold on different online websites), Smith and Brynjolfsson conclude that higher prices on well-known websites do not affect the sales of products because of the brand effect carried by the well-known websites [116]. Erdem and Keane performed a temporal analysis of brand effects and found that advertising intensity has only weak short run effects on brand adoption, but has a strong cumulative effect in the long run [35].

All these studies suggest that brand does play an important role in user item adoptions. However, there are very few works focusing on modeling brand effect in item recommendation and/or adoption [56, 125, 138]. In [138], researchers studied the correlation between the brands liked by a social media user and his items purchased to make recommendations of items of new brands. This work focuses on user brand preference but overlooks the social network information. The work in [56] performed user classification to find the so-called “brand-sensitive users”, e.g., those who adopt mostly items from brand names. This concept of brand-sensitive users bears some similarity with the concept of “brand-conscious users” in our models in Chapter 3. However, their method requires an item taxonomy and combines the taxonomy with the user classification to predicts adoption of *item-brand combinations*. Our method does not require the availability of the taxonomy as we learn the semantic grouping, i.e., topic of items directly from data and we predict adoption of items instead of item-brand combinations.

Table 2.1: A taxonomy of models for adoption dynamics. Places where no model can be found are marked with "None".

Data Entities	Dynamicity of Data	**Social Network Consideration**	
		No	Yes
(Users, Items)	*Static*	Probabilistic MF [96], Non-negative MF [74], Latent Dirichlet Allocation [13]	Social Recommender [85], Social Trust Ensemble [86] Topic model with Network [95] Unified Generative Model [22]
	Dynamic	Dynamic MF [23], Dynamic topic model [12]	Topic Interaction & Homophily Aware Model (Chapter 5), Cooperation & Competition [98]
(Users, Items) + Authors	*Static*	Author-Topic-Model [119]	Topic model with Network [95]
	Dynamic	Temporal-Author-Topic [26]	None
(Users, Items) + Brands	*Static*	Brand-Item-Topic (BIT, Chapter 3)	Social BIT (Chapter 4)
	Dynamic	None	None
(Users, Items) + Others	*Static*	HYbrid REcommender System [59], MF + Side Information [4, 101]	None
	Dynamic	Temporal Recommender with Item Taxonomy [66]	None

2.2 A Taxonomy of Models

We provide a taxonomy of models for adoption dynamics in Table 2.1. The taxonomy plays the role of the dissertation's roadmap connecting various models and also highlights where our contributions fit into the big picture. In the taxonomy, we classify all models by three criteria, namely:

- *Social Network*: This separates models incorporating social network from those not incorporating social network.

- *Data entities* : This refers to the data entities associated with the users and items contributing to adoptions and ratings. The typical data entities are user-item pairs representing adoption/rating data. Other entities are then augmented to enirch the model. These additional entities can be (i) authors, e.g. in the context of paper citation, (ii) brands, or (iii) others e.g. comments, reviews or meta data of adopted items.

- *Dynamicity* of data: : This refers to the modeling of temporal aspect of the data. When time is not part of the model, we say the data modeled is static. Otherwise, the data is said to be dynamic.

Models without Considering Social Network

There are many works on modeling adoptions without considering social network. Under this class of models, most focus on modeling user-item ratings or adoptions only. They typically reduce the high dimensionality of user-item matrix either by (i) factorizing it into two low-rank matrices or (ii) finding semantic grouping of items into "topics" and learning user interests in such "topics". Models of the first family are thus called matrix factorization models, e.g. Probabilistic Matrix Factorization (PMF) [96] and Non-negative Matrix Factorization (NMF) [55]. Meanwhile the second family includes topic models such as Latent Dirichlet Allocation (LDA) [13, 50]. However, both families originally only deal with *static* data. To handle *dynamic* data due to shift in user interests over time, Chua *et al.* incorporated dynamical systems theory into NMF and proposed Dynamic Matrix Factorization (DMF) in [23]. Meanwhile, Blei *et al.* proposed a dynamic version of LDA in [12].

Beyond the basic models, several works propose to incorporate more entities to exploit more information and structure in data. For example, the model Author-Topic-Model (ATM) [110, 119] incorporates authors and thus can learn their topic interests. ATM is later extended to incorporate dynamic data in the model known as Temporal-Author-Topic (TAT) [26] to deal with dynamic data.

Among the models incorporating data entities, there are few of them that incorporate brand entity. New factors can thus be associated with brands to more accurately model the item adoption behavior. In this thesis, we thus fill the gap by proposing a model called Brand-Item-Topic (BIT) [84] which infer *brand-consciousness* of user and *exclusiveness* of brands. We also develop its distributed version, abbreviated as DeBIT, to handle large-scale data. Both BIT and DeBIT are presented in Chapter 3.

Finally, there are models which incorporate other data entities such as comments, reviews or meta-data of items. For instance, (i) HYRES [59] incorporates item comments/reviews to mine consumer sentiment, (ii) [101] and [4] propose MF models which incorporate meta-data of adopted items as side information, and (iii)

[66] proposes to incorporate item taxonomy and associated bias into recommendation systems.

Models considering Social Network

As shown in Table 2.1, there are several models incorporating social network information. First, we describe those that deal with static adoption data. Among them, we have extensions of MF and LDA to incorporate social network information e.g. trust, friendship or follow links into *static* adoption data. Examples of MF extensions are Social Recommendation (SoRec) [85], and recommendation with Social Trust Ensemble (STE) [86]. Inspired by the success of these social recommendation models, we also incorporate social network data into our BIT models and propose a model called SocBIT in Chapter 4.

Examples of topic model extensions include a topic model with network regularizer [95], a topic model for role discovery [93], and Unified Generative Model (UGM) [22]. The first two combine topic modelling with author network analysis to discover topics and topical communities. The third work exploits the social network to learn the extent to which a user relies on their social relationships to make adoption decisions.

To handle *dynamic* data, we propose a model called Topic-level Interaction and Homophily Aware Diffusion (TIHAD) [83] to handle diffusion under the effect of homophily and item interaction. A recently proposed model called IMM [98] also deals with information diffusion under cooperation and competition effect but it does not incorporate homophily.

In the next few sections, we will elaborate some of the above-mentioned models in greater detail.

2.3 Matrix Factorization Models

Matrix factorization (MF) based recommendation methods have been shown to yield accurate results in recent years [8, 51, 67, 68, 71, 74, 96, 100, 107, 111, 120, 137]. The underlying assumptions of these methods can be described as follows. A user may adopt/rate lots of items, say thousands of items, however, his/her *preference* should be summarizable by a much smaller number of factors. Similarly, an item can have a large number of features, however, there should be a compact representation for the item using just a small number of essential factors. In other words, it is assumed that there is a low dimensional space of *latent* factors in which each user u and item i can be represented as vectors θ_u and θ_i respectively. Moreover, the proximity of two vectors θ_u and θ_i represent the similarity of u and i, which in turn can be used to approximate the amount of interest (or rating) of u toward i.

In technical terms, these methods take an observed $N \times M$ user-item rating matrix $\mathbf{R}$ and find a factorization of it into two low-rank matrices Θ_U and Θ_I of rank K much smaller than number N of users and number M of items. Each row in matrix Θ_U represents a vector of latent factors of a specific user u and each column in matrix Θ_I represents a vector of latent factors of a specific item i. Mathematically, traditional MF methods search for matrices Θ_U and Θ_I which minimize the squared error:

$$||\mathbf{R} - \Theta_U^T \Theta_I||_F^2 = \sum_{u,i} (r_{u,i} - \theta_u^T \theta_i)^2 \tag{2.1}$$

where $||.||_F$ denotes the usual Frobenius norm of matrix.

A well-known issue of traditional MF methods is the interpretability of inferred latent factors. To fix this issue, a variant, called non-negative matrix factorization (NMF), was proposed in [70, 71], originally for modeling image pixels. To guarantee non-negativity, the update algorithm of NMF has a multiplicative form instead of substractive form in MF. The non-negative latent factors returned by NMF can then be interpreted as user topic interest as well as item topic features. The success of NMF inspires more extensions with applications such as document clustering in

[32, 33, 113, 126, 132], handling data sparseness in [54, 55, 78].

Another variant is the Probabilistic Matrix Factorization (PMF), introduced in [96] by Salakhutdinov and Mnih. The authors later extended it to Bayesian Probabilistic Matrix Factorization (BPMF) by providing a full Bayesian treatment through Markov Chain Monte Carlo (MCMC) algorithm [111]. These methods produce good predictive accuracy and can handle *cold start* problem, e.g. predicting future ratings for users with few or no observed ratings. The contribution of PMF is the addition of a Gaussian noise

$$\varepsilon \sim \mathscr{N}(0, \sigma^2)$$

such that the probability of observing $r_{u,i}$ is now given by

$$p(r_{u,i} | \theta_u, \theta_i, \varepsilon) = \mathscr{N}(\theta_u^T \theta_i, \sigma^2)$$

PMF obtains its parameters by maximizing a (log) posterior using stachastic gradient descent while BPMF adds additional Gaussian and Wishart priors to use Gibbs Sampling approach.

In real life, recommendation and adoption processes usually involve social influence among socially connected users — the connections may be friendships, trusts, follow links or others. However, traditional MF approaches used in Recommender System (RS) applications do not consider such social influence. Thus, several recent works [9, 60, 85, 86, 89, 91] propose to incorporate such social influence by employing the social network among users. As demonstrated in the works, augmenting the social network information, representable as a user-user social weight matrix, indeed increases the recommendation accuracy remarkably.

A representative among these works is a PMF extension called SoRec [85] which factorizes simultaneously the observed *rating matrix* **R** and the *user-user social weight matrix* **W** into user and item latent factors as shown below:

$$\mathbf{W} \approx \blacksquare_U^T \mathbf{Z} \text{ and } \mathbf{R} \approx \blacksquare_U^T \blacksquare_I \tag{2.2}$$

where $\blacksquare_U$ and $\blacksquare_I$ are user and item latent factor matrices as in traditional matrix factorization. $\mathbf{Z}$ is another user latent factor matrix for generating the social weight matrix. However, introducing a second user latent factor matrix $\mathbf{Z}$ reduces the interpretability of SoRec model. To address the interpretability issue, two other variants of SoRec model were proposed in [86] and [89] respectively. The first, called *recommendation by Social Trust Ensemble* (STE), assumes that neighbors of a user directly influence his *ratings* instead of his latent factors. The second variant, called *Recommendation with Social Regularization*, employs latent factors of a user's neighbors to regularize the latent factors of the user himself. The next work in this direction [60] proposed to learn latent factors of a user as a weighted average of his neighbor features, where the weights are the trust values among users. The proposed model was demonstrated to outperform SoRec and STE. On the whole, all these models extend the traditional matrix factorization approach by augmenting the social network information. However, none of them consider brand-related factors.

As there is a hierachical structure {topics → brands → items} in our BIT model, it is interesting to note some similarities and differences between our BIT and recent works [66, 114] employing hierachical matrix factorization. [66] is a model for music recommendation and it has the following differences with ours.

- The hierachy is employed to handle *sparsity* as items of the same ancestor in the taxonomy can share certain information, e.g., tracks of the same artists/album can share similar ratings.
- The model incorporates *biases* such as (i) rating order bias, i.e., the tendency that users rate items in the context of previous items they rated, and (ii) bias toward popular artists. The latter creates a "long tail" effect for less popular artists, who are either mediocre or exclusive. Although the exclusive artists are similar to exclusive brands in ours, the model in [66] does not distinguish exclusive from mediocre ones.
- It does not consider and capture user brand consciousness.

Meanwhile, authors of [114] consider the problem of factorizing a **plant** × **trait** matrix. They exploit the hierachy **phylogenetic group**, **family**, **genus**, **species**, and **plant** and assume that the observed **plant** × **trait** matrix is obtained as the lowest matrix in a hierachy of matrices such as **species** × **trait** matrix, **genus** × **trait** matrix and so on. Applying this to adoption scenario creates a hierachy of the following matrices: **user** × **topic**, **user** × **brand** and **user** × **item**. Although adding the **user** × **brand** matrix is somewhat different from our BIT models, it is interesting to study the relationship between this approach and ours in the future work. For instance, the **user** × **brand** matrix may be used to represent users' preference for brands.

2.4 Topic Models

Closely related to NMF is the family of probabilistic topic models, which were originally built for automatic topic discovery in corpora of documents. A well-known representative of this family is the Latent Dirichlet Allocation (LDA) model proposed by Blei *et al.* [13]. The success of LDA in automatic topic discovery has inspired a variety of its adaptations to various tasks in corpus summary and notations, e.g. author topic interest detection [109, 110, 119], (real-time) topic detection for streaming text data [30, 103, 127, 128, 131, 133], developer contribution evaluation based on data from code repository [77] and so on. We now briefly review LDA and a variant of it which has some similarity with our work, the so-called Author-Topic model by Rosen *et al.* [109, 110, 119].

Latent Dirichlet Allocation. Latent Dirichlet Allocation (LDA) was proposed by Blei in his founding paper [13] as a generative model which simulates how documents in a corpus are generated. The model represents *each topic as a distribution over words* and posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. The whole process can be represented by the graphical model in Figure 2.1. The generative process is as follows.

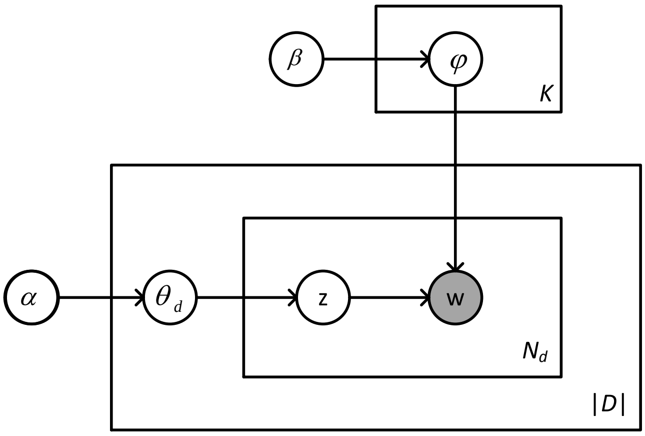


Figure 2.1: Plate diagram of LDA

1. The set of topics of all documents in the corpus are assumed to have exactly K topics. Each topic $k \in [1,K]$ is represented as a distribution φ_k over words in a given dictionary. All φ_k's are sampled from a Dirichlet distribution with hyper parameter β:
$$\varphi_k \sim Dirichlet(\beta)$$
2. Each document d is associated with a topic distribution θ_d indicating which topics are being discussed in d. θ_d is drawn from the following Dirichlet distribution with hyper parameter α
$$\theta_d \sim Dirichlet(\alpha)$$
3. Each word w in d is then generated by the following two steps. Let us assume that w is the word at i-th position in d, thus we denote it as $w_{i,d}$.
   (a) sampling a topic $z_{i,d}$ for it from the topic distribution θ_d:
   $$z_{i,d} \sim Multinomial(\theta_d)$$
   (b) given that topic $z_{i,d} = k$ for some $k \in [1,K]$, $w_{i,d}$ is sampled from the

word distribution φ_k:

$$w_{i,d} \sim Multinomial(\varphi_k)$$

Inferring parameters of LDA is basically maximizing a likelihood subject to probabilistic constraints. Blei *et al.* showed that the optimization can be solved by variational expectation maximization [13] while Griffiths and Steyvers subsequently showed that LDA inference can be performed easily using Gibbs Sampling [46].

LDA can be adapted to model a set of adoptions as follows. Each user is considered as a "document" and his topic interest corresponds to the topic distribution of the document. Each item adopted by the user is then considered as a "word" sampled from one of those topics. For example, a user plans to buy some movie DVDs and he is interested in topics/genres "action", "science fiction" and "comedy" with preference levels 0.5, 0.25 and 0.25 respectively. He thus looks at movies under "action" and may buy some action movie(s). He may also buy some movie(s) under "science fiction" and "comedy" as well. However, this adaptation of LDA only models user adopting item based on topic — as the only cause of adoption — without considering other important factors, especially brand-related factors such as brand of the item and whether the user relies on item brand to make his final decision.

Another limitation of this LDA variant is the fact that it ignores the social relationships among users. Hence, several authors [7, 21, 22] have extended LDA to relate the user-user relationships with user-item adoptions. Balasubramanyan and Cohen [7] proposed Block-LDA which unifies the Mixed Membership Stochastic Blockmodels [5] and LDA to incorporate the user-user relationships. The Mixed Membership Stochastic Blockmodels by Airoldi *et al.* uses probability distributions to represent the fact that a user can belong to a set of communities, each with varying degree of memberships. Building on this, Block-LDA jointly models the sparse relationships between users and the co-occurence of users and items in documents.

Specifically, text documents may contain words which refer to both user names and items; users who co-occur in the same document are considered as socially related. Different from Block-LDA, in models proposed by Freddy *et al.* [21, 22], users are the documents who adopt items modeled as words in LDA. Users do not adopt other users and items do not have links between them.

Author-Topic Model. Using LDA for modeling and querying information from corpora of documents is restrained by a major limitation — it does not include *authorship* information. To overcome this, Rosen-Zvi *et al.* extended LDA to the Author-Topic Model (ATM) in [109, 110, 119]. ATM also aims at learning *topic interests of authors*. Basically, ATM jointly models the content of documents and the topic interests of authors. The model associates each author with a multinomial distribution over topics which represents his/her interest. However, the model assumes each word in a document comes only from one author, who generates topics independently from other authors. This may not be true in some contexts (i) research collaboration where authors of a paper coordinate with each other to generate topics of the paper; (ii) item adoption with presence of brands as an item can be jointly created by several brands as a result of the cooperation of the brands. Nevertheless, ATM is shown to perform successfully in topic and author assignment tasks to new documents, especially for scientific corpora such as NIPS or CiteSeer. The success motivates other variants of ATM for learning contributions of software developers [77] or topic interest of twitter users [133].

Neither matrix factorization nor topic models consider *brand effects* that may affect adoption and/or the generation of user-item ratings. Actually, there are very few works focusing on modeling brand effect on ratings and item adoptions [56, 125, 138].

Another important aspect of adoption dynamics is the *time* dimension. All reviewed models, however, do not involve time. Thus, they are not appropriate for modeling diffusion of items. Moreover, in real-world, items are not diffused/adopted in isolation from others. More than often, diffusion of one item can trigger or impede diffusion of another. We thus review a class of models for diffusion processes with presence of interaction among items being diffused.

2.5 Diffusion Models Considering Item Interaction

It is noteworthy that most existing models of diffusion are built upon *independent* contagion assumption whereby the diffusion of each item is assumed (at least implicitly) to happen independently from other items. The interaction among items during diffusion is thus left out of the picture. To handle this issue, authors in [11, 63, 102] adapt the dynamical systems theory for diffusion of species to model diffusion of two *competing* products or viruses in a network. In this approach, diffusion processes are modeled as solutions of some system of partial differential equations for predator and prey species [135]. However, when there are more than two items, such systems get complicated very fast and solving them becomes so difficult or even impossible. In fact, it has been shown that even with only three species in the system, the dynamics of the system is already very complicated with peculiarities [92]. Thus, most works by this approach are limited to the case of two items.

Compared with the above state of the art, our work in Chapter 5 manages to model diffusion processes of an arbitrary number of items by tackling the problem from a different approach. By capturing interaction between any two items as the (dis)similarity between them with respect to a number of latent factors, our approach can formulate a model of which parameters can be learned as solution of a constrained optimization problem, which in turn can be solved using Projected Gradient Descent. Interestingly, there is another work by Myers and Leskovec [98] which is also able to model diffusion of an arbitrary number of interacting items. The first major difference with our work is that their model actually model interactions of item clusters — e.g. items under the same topic — rather than pairwise item interactions. Moreover, our model is also capable of modeling homophily effect, which is not considered in [98].

2.6 Scalable Algorithms

To apply a Machine Learning (ML) algorithm to large scale datasets, for which a single machines computational power is inadequate, a common strategy is *data parallelism* by partitioning data across machines. Each machine is then allowed to read and update all parameters of the model. Data parallelism however faces two issues: (i) parameters may be dependent, thus a naive way of concurrently updating can introduce errors that slow down convergence or even cause algorithm failure, and (ii) model parameters converge at different rates, thus a small subset of parameters can bottleneck the whole ML algorithm. To address the two issues, an alternative and complimentary strategy, namely *model parallelism*, has been proposed, see e.g., [72, 80, 136]. As suggested by the name, model parallelism partitions the model parameters for non-shared parallel access and updates.

While the *progress* of a conventional computer program can be measured by *throughput* (operations per unit time), that of an ML program is measured by a numerically explicit objective function specified by the ML application. More progress means the objective increases (or decreases) to approach an optimum at a faster rate. It is important to note that *progress per iteration* is distinct from *iteration throughput* (number of iterations executed per unit time); effective ML implementations combine high progress per iteration with high iteration throughput, yielding high progress per unit time. In Chapter 3, we measure iteration throughput against the number of machines to evaluate the efficiency and scalability of our Distributed and enhanced BIT (DeBIT) model. More importantly, for the implementation of DeBIT, we chose the approach of *parameter server* [49, 73] which combines both data parallelism and model parallelism.

A natural concern arises from using parallel algorithms is on the correctness of such algorithms. Although this concern was already addressed for gradient-based methods [14, 105, 139], a theoritical guarantee for that of parallel Markov Chain Monte Carlo (MCMC) methods, including parallel Gibbs sampling, has not yet existed. Nevertheless, various experimental works demonstrated that parallel Gibbs sampling provide good performance [64, 99, 117, 136]. Most recently, authors of [64] show that it is possible to cut the sampling error to a very small level. This suggests us that the distributed algorithm of our DeBIT has at least an empirical guarantee for its correctness. This is verified by our experiments for DeBIT's accuracy in Chapter 3.

Chapter 3

Models for Brand Effects in Item Adoption

3.1 Introduction

In this chapter, we focus on modeling user-item adoptions that can be attributed to (i) user topic and brand factors; and (ii) item topic and brand factors. It is noteworthy that two concepts “topic” and “brand” are interpreted in a broad sense. A “topic” refers to a semantic grouping of items while a “brand” refers to a person, trademark, or business that produces items. For example, the topic of a KFC restaurant is fast food while KFC is a brand. In the case of movie, brand may refer to the director and lead actors in a movie.

As shown in Section 2.1, brands have remarkable effect on user-item adoption. Thus, we first identify brand-related factors which motivate adoption behavior. By careful analysis of literature, we found two such factors, namely *user brand consciousness* and *brand exclusiveness*. Examples of exclusive brands include elite universities, luxury fashion brands and cars. An example of a brand-conscious user is one who enjoys driving a luxury car, wears designer clothes, and go for a fine dining restaurant. In general, brand consciousness of a user measures the tendency/degree that the user adopts items from exclusive brands and an exclusive brand is one with a high probability of being chosen by brand-conscious users. Thus, we model a users brand consciousness as the probability of the user adopting items from exclusive brands. Henceforth, in this work, we measure user brand consciousness and brand exclusiveness by numeric values instead of binary values. The numeric values are more informative than the binary values as they allow us not only to discover exclusive brands and brand-conscious users but also to rank such brands and users by their exclusiveness and brand consciousness levels respectively.

We also would like to clarify that exclusive brands are not limited to expensive brands. For example, elite schools are exclusive. Gaining admission to these schools is not easy even when their fees are not necessarily high. Publishers can also be exclusive when they maintain very high standards in their collections of publications. However, they may require low publishing fees and subscription fees as a way to attract good quality publications and to reach out to many libraries.

Research Problem and Contributions.

To incorporate the brand-related factors, we propose a generative model called **Brand-Item-Topic Model** (**BIT**). BIT extends the well known Latent Dirichlet Allocation (LDA) model [13] by modeling brands and decisions of item adoptions, in addition to modeling the latent topics of users and items. The major goal is thus to discover latent variables (topic, brand and decision) from the observed item adoption data.

One may argue that brand(s) of a given item are already observed, so BIT should not consider brands as latent variables. In fact, for an item co-created by multiple brands, e.g. a research paper created by multiple co-authors, a movie casted by several popular actors/actresses, only a small subset of the brands are considered in a user's adoption decision. In many cases, only one of the item's brands actually leads to such an adoption decision. Examples include (i) movie adoption due to presence of one favorite actor/actress, (ii) paper citation due to presence of one outstanding, well known author. Thus, in this work, we assume that she relies on *only one* brand for the decision and we usually do not observe this brand. Generally, for an item associated with multiple brands, it is interesting but challenging to discover the actual *latent brand* from which a user chooses to adopt the item. We design BIT model such that it will recover such actual latent brands in brand-based adoption decisions. In the case of single-brand item, although the brand is observable, we still do not know whether the user actually relies on the brand to make adoption decision. Thus, it is still interesting to model the process of adoption decision using brand-related latent variables.

BIT, to the best of our knowledge, is the first topic model that explicitly examines the brand-related factors in adoption dynamics. There are several research challenges in formulating this model. Firstly, we do not assume (i) item price information is available, or (ii) exclusive brands produce expensive items. We thus cannot determine *exclusive* brands simply by their high price items compared with the price of similar items under other brands. Not relying on price information however inspires us to design a model that can be applied even when neither assumptions are true. Not determining exclusive brands purely based on price information also matches with studies from marketing literature, where brands are evaluated by *brand equity* i.e. the incremental utility with which a brand endows a product, compared to its non-branded counterpart [37].

Secondly, it is not trivial to evaluate models that infer the brand-related factors from item adoption data due to a lack of ground truth data. It is possible to solicit user input about their item adoption decisions but this evaluation approach has several drawbacks. It is clearly not scalable. Either users find it intrusive or they may not recall their adoption decisions. We thus have to evaluate BIT using alternative approaches.

The first part of this chapter (Sections 3.2 to 3.4.4) addresses the challenges by offering the following contributions.

- We propose a novel topic model BIT for inferring brand and topic latent variables that generate a set of observed item adoptions without price information. The model introduces a richer structure of adoption process.

- To evaluate BIT, we generate a synthetic dataset where *exclusive* brands and *brand conscious* users are injected and controlled by a set of parameters. We show that BIT outperforms baselines in learning the ground truth variables and it also achieves reasonable accuracy in recovering brand conscious users and exclusive brands.

- We also evaluate BIT using two real datasets from FourSquare and ACM Digital Library. The exclusive food outlet brands learnt from the Foursquare data by BIT are shown to be more pricey than the non-exclusive ones. We also show that the exclusive authors learnt from the latter dataset have higher h-index than those non-exclusive ones.

3.2 Brand-Item-Topic (BIT) Model

3.2.1 Latent Dirichlet Allocation and Our Model

Before elaborating our Brand-Item-Topic Model, we would like to briefly present a well-known topic model, Latent Dirichlet Allocation (LDA) [13]. LDA is a generative Bayesian model in which each document of a collection is modelled as a finite mixture over an underlying set of topics. Each document is associated with a *topic distribution*. Each topic is in turn modeled as a *distribution over terms*. In the context of item adoption, LDA can be adapted as follows.

- Each topic can be considered as a *category* of items. Thus, a topic is characterized by a distribution over *items*.

- Each user is modelled as a "document" e.g. we can model his/her preference as a topic distribution. More precisely, the adoption history of a user can be considered as a "document of adopted items" where each item is generated under some favorite topic of the user.

Our proposed model, Brand-Item-Topic Model (BIT), incorporates brand preference into the topic model. To choose an item for adoption, a user first chooses a topic z that she is interested in. Then the user chooses the item to adopt based on either one of the following methods:

1. *Topic-based Adoption*: With the chosen topic z, the user adopts one of the many items under z. For example, when the user buys a novel, a topic-based adoption will have the user first selects a genre (e.g. sci-fi) among his topic preferences and selects a novel based on its popularity under the genre.

2. *Brand-based Adoption*: The user chooses a brand b from topic z, then selects an item from the item distribution of brand b. For example, when user buys a novel, the user first selects a genre, then a preferred author under the selected genre, follow by selecting a novel written by the preferred author.

It can be seen that for a given user, the choice of method for adoption reveals the importance of brand to her. If she usually prefers to adopt based on popular brands, we may say that she has a *brand-preference*. Understanding users' brand preferences allows us to utilize more information for making better recommendations to them.

3.2.2 Generative Process

We illustrate the generative process of our Brand-Item-Topic Model (BIT) using the graphical model in Figure 3.1a and its notations in Table 3.1. The process has two following stages.

1. *Generating distributions from priors*:

   (a) Multinomial topic and decision distributions of each user u are sampled respectively, $\vartheta_u \sim Dir(.|\theta)$ and $\delta_u \sim Dir(.|\gamma)$.

   (b) Item and brand distributions of each topic k are sampled respectively, $\varphi_k \sim Dir(.|\phi)$ and $\psi_k \sim Dir(.|\alpha)$.

   (c) For each brand b, we sample an item distribution $\omega_b \sim Dir(.|\beta)$ from Dirichlet distribution with prior parameter β.

Table 3.1: Notations of BIT

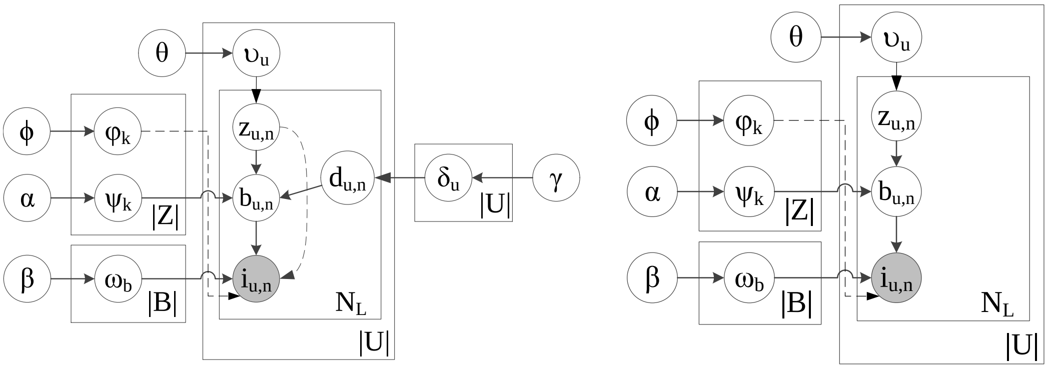
Notation	Description
$i_{u,n}$	Item at n-th adoption of user u
$z_{u,n}$	Latent topic of $i_{u,n}$
$b_{u,n}$	Latent brand of $i_{u,n}$
$d_{u,n}$	Latent decision variable of this adoption
ω_b	Parameters for the item distribution of brand b
β	Hyperparameter for Dirichlet prior of ω_b
ψ_k	Parameters for the brand distribution of topic k
α	Hyperparameters for Dirichlet prior of ψ_k
ϑ_u	Parameters for the topic distribution of user u
θ	Hyperparameters for Dirichlet prior of ϑ_u
φ_z	Parameters for the title distribution of topic z
ϕ	Hyper parameters for Dirichlet prior of φ_z's
δ_u	Parameters for binomial distribution of $d_{u,n}$
γ	Hyper parameter for Dirichlet prior of δ_u's
U and N	Set and number of users
Z and K	Set and number of topics
B and Q	Set and number of brands
I and M	Set and number of items

2. *Generating adoptions*:

   (a) User u makes n-th adoption by first sampling a topic $z_{u,n} = k$ from her topic distribution ϑ_u, i.e. $z_{u,n} = k \sim Multi(\vartheta_u)$.

   (b) She then decides to rely on either topic or brand to pick her adoption. This decision is realized by a flag $d_{u,n} \sim Bernoulli(\delta_u)$.

   (c) If she decides to rely on topic then the item will be picked from topic.



(a) Brand-Item-Topic Model (BIT)

(b) Simplified Brand-Item-Topic Model (sBIT)

Figure 3.1: Bayesian network for BIT and sBIT

Otherwise, she picks a brand and then an item under it.

$$\text{If } d_{u,n} = \mathtt{T} \text{ then } i_{u,n} \sim \ \varphi_k \tag{3.1a}$$

$$\text{If } d_{u,n} = \mathtt{B} \text{ then } b \sim Multi(\psi_k) \text{ and } i_{u,n} \sim \ \omega_b \tag{3.1b}$$

We would like to note some important features of our model.

- All Dirichlet priors are symmetric.
- The brand-item distribution w_b is not observable from data since the brand b may be *latent*.

It can also be seen that when all adoption decisions are topic-based, the BIT generative process degenerates to that of LDA. Hence, LDA can be considered as a special case of BIT. On the other hand, if all decisions are brand-based, we have a simplified version of BIT, denoted as sBIT (see Figure 3.1b), which later will be used as another baseline in evaluating BIT. Although sBIT is related to LDA by the equality $\sum_b p(i|b)p(b|z) = p(i|z)$, the two models are still significantly different. In fact, although LDA can infer the values $p(i|z)$, it cannot go further to derive the values $p(i|b)$ and $p(b|z)$, which are item popularity under a given brand and brand exclusiveness/popularity under a given topic.

Training BIT requires extensive computation cost as we have five distributions to be learned. Thus, instead of using variational methods [13], [50], and the Expectation-Maximization algorithm [29], [106] we adopt Gibbs sampling [41], [16].

3.3 Training BIT

Given hyper parameters $\Pi = (\alpha, \beta, \gamma, \delta, \phi)$, the training process of BIT involves two tasks (i) learning latent variables, and (ii) learning distributions. We present our solutions for two tasks in Sections 3.3.1 and 3.3.2 respectively. The notations for the training process are summarized in Table 3.2.

3.3.1 Learning Latent Variables

Using the approach of collapsed Gibbs sampling, we can learn latent topics, brands and decisions by an *alternating* update process. We start from an initial guess for the variables and repeat the following update process until convergence.

- Using *current* values of latent decisions $\mathscr{D}_c$, latent topics $\mathscr{Z}_c$ and latent brands $\mathscr{B}_c$, we sample *new* latent decisions $\mathscr{D}_n$ based on Prop. 1.
- Using $\mathscr{D}_n, \mathscr{Z}_c, \mathscr{B}_c$, we sample *new* latent topics $\mathscr{Z}_n$ based on Prop. 2.
- Using $\mathscr{D}_n, \mathscr{Z}_n, \mathscr{B}_c$, we sample *new* latent brands $\mathscr{B}_n$ based on Prop. 3.

Following are propositions for learning the three sets of latent variables: topic, brand and decision variables (for detailed proof see Appendix A). The main idea here is *alternating* update of each variable assuming all remaining ones are known. We start from an initial guess and repeat the corresponding update processes until convergence.

Note that in all the propositions, we use shortened index $j = (\tilde{u}, \tilde{n})$ to denote a specific $\tilde{n}$-th adoption of a given user $\tilde{u}$. The specific item that $\tilde{u}$ adopts at j-th adoption is denoted as $\tilde{i}$ and the set of brands of $\tilde{i}$ is $B_{\tilde{i}}$. Vice versa, $I_{\tilde{b}}$ is the set of items of brand $\tilde{b}$. Finally, vectors are conventionally represented in boldface. We provide a summary of notations used in the propositions in Table 3.2.

Proposition 1 (Updating latent decisions)**.** *Given the* ***current*** *assignment* $\tilde{z}$ *for topic variable* z_j *and the* ***current*** *assignment for brand variable* b_j*, we sample* ***new*** *latent decision* d_j *as follows.*

1. *When* $b_j = -1$*, there is no brand assigned to adoption* j*. Thus, we have*

$$P(d_j = \tilde{d} | \mathscr{D}_{-j}, \mathscr{I}, \mathscr{Z}, \mathscr{B}; \Pi) \propto \begin{cases} \dfrac{\gamma + dc_{0,\tilde{u}} - 1}{2\gamma + n_{\tilde{u}} - 1} \cdot \dfrac{1}{M} \cdot \dfrac{|B_{\tilde{i}}|}{Q}, & if\, \tilde{d} = 0 \\ \dfrac{\gamma + dc_{1,\tilde{u}} - 1}{2\gamma + n_{\tilde{u}} - 1} \cdot \dfrac{\phi + c_{\tilde{z},\tilde{i}} - 1}{M\phi + \sum c_{\tilde{z},i} - 1}, & if\, \tilde{d} = 1 \end{cases} \tag{3.2}$$

Notation	Description
n_u	Number of adoptions of user u
$i_u = \{i_{u,1}, \dots, i_{u,n_u}\}$	Vector of adoptions by user u
$z_u = \{z_{u,1}, \dots, z_{u,n_u}\}$	Corresponding latent topics
$b_u = \{b_{u,1}, \dots, b_{u,n_u}\}$	Corresponding latent brands
$d_u = \{d_{u,1}, \dots, d_{u,n_u}\}$	Corresponding latent decisions
$\mathscr{I} = \{i_1, \dots, i_N\}$	Vector of item adoptions by all users
$\mathscr{Z} = \{z_1, \dots, z_N\}$	Vector of latent topics of adoptions.
$\mathscr{B} = \{b_1, \dots, b_N\}$	Vector of latent brands of adoptions.
$\mathscr{D} = \{d_1, \dots, d_N\}$	Vector of latent decisions of adoptions
$\mathbf{c}_u = (c_{u,1}, \dots, c_{u,K})$	Topic counts for a user u
$\mathbf{c}_z = (c_{z,1}\ , \dots, c_{z,M})$	Item counts for adoptions by topic z
$\mathbf{c}_b = (c_{b,1}\ , \dots, c_{b,M})$	Item counts for brand-based adoptions by brand b
$\mathbf{bc}_z = (bc_{z,1}, \dots, bc_{z,Q})$	Brand counts for topic z
$\mathbf{dc}_u = (dc_{u,d=\mathtt{T}}, dc_{u,d=\mathtt{B}})$	Decision counts for u

Table 3.2: Notations used in inference

2. *When $b_j = \tilde{b}$, brand $\tilde{b}$ is assigned to adoption j. Thus, we have*

$$P(d_j = \tilde{d} | \mathscr{D}_{-j}, \mathscr{I}, \mathscr{Z}, \mathscr{B}; \Pi) \propto \begin{cases} \dfrac{\gamma + dc_{0,\tilde{u}} - 1}{2\gamma + n_{\tilde{u}} - 1} \cdot \dfrac{\beta + c_{\tilde{b},\tilde{i}} - 1}{|I_{\tilde{b}}|\beta + \sum c_{\tilde{b},i} - 1}, & if\, \tilde{d} = 0 \\ \dfrac{\gamma + dc_{1,\tilde{u}} - 1}{2\gamma + n_{\tilde{u}} - 1} \cdot \dfrac{1}{M}, & if\, \tilde{d} = 1 \end{cases} \tag{3.3}$$

Proposition 2 (Updating latent topics)**.** *Given the **current** assignment $\tilde{d}$ for decision variable d_j and the **current** assignment $\tilde{b}$ for brand variable b_j, we can sample a **new** assignment for latent topic z_j using Equation 3.4.*

$$P(z_j = \tilde{z} | \mathscr{Z}_{-j}, \mathscr{I}, \mathscr{B}, \mathscr{D}; \Pi) \propto \begin{cases} \dfrac{\phi + c_{\tilde{z},\tilde{i}} - 1}{M\phi + \sum c_{\tilde{z},i} - 1}, & if\, \tilde{d} = 1 \\ \dfrac{\alpha + bc_{\tilde{z},\tilde{b}} - 1}{|B_{\tilde{i}}|\alpha + \sum bc_{\tilde{z},b} - 1} \cdot \dfrac{\beta + c_{\tilde{b},\tilde{i}} - 1}{|I_{\tilde{b}}|\beta + \sum c_{\tilde{b},i} - 1}, & if\, \tilde{d} = 0 \end{cases} \tag{3.4}$$

Proposition 3 (Updating latent brands)**.** *Given that $i_j = \tilde{i}$, the **new** assignment $\tilde{d}$ for decision variable d_j and the **new** assignment $\tilde{z}$ for topic variable z_j, we can sample a **new** assignment for latent brand b_j using Equation 3.5.*

$$P(b_j = \tilde{b} | \mathscr{B}_{-j}, \mathscr{I}, \mathscr{Z}, \mathscr{D}; \Pi) \propto \begin{cases} 0, & if\, \tilde{d} = 1, \tilde{b} \neq -1 \\ 1, & if\, \tilde{d} = 1, \tilde{b} = -1 \\ \dfrac{\alpha + bc_{\tilde{z},\tilde{b}} - 1}{|B_{\tilde{i}}|\alpha + \sum bc_{\tilde{z},b} - 1} \cdot \dfrac{\beta + c_{\tilde{b},\tilde{i}} - 1}{|I_{\tilde{b}}|\beta + \sum c_{\tilde{b},i} - 1}, & if\, \tilde{d} = 0 \end{cases} \tag{3.5}$$

3.3.2 Learning Distributions

Once we have learned all latent variables, they can be used to estimate five distributions $\vartheta_u, \delta_u, \psi_k, \varphi_k, \omega_b$ which we are interested in. Similar to LDA, the conjugacy of Dirichlet and Multinomial distributions can be used to show that all the parameters $\vartheta_u, \delta_u, \psi_k, \varphi_k, \omega_b$ follow Dirichlet posteriors.

Proposition 4 (Learning distributions)**.** *The five interested distributions can be learned as follows.*

1. *Given a user u, his/her topic distribution ϑ_u and decision distribution δ_u follow Dirichlet posteriors parameterized by $\theta 1 + \mathbf{c}_u$ and $\gamma 1 + \mathbf{dc}_u$ respectively. Thus, we have:*

$$P(\vartheta_u|\mathscr{Z}, \theta) = Dir(\vartheta_u|\theta 1 + \mathbf{c}_u) \tag{3.6}$$

$$P(\delta_u|\mathscr{D}, \gamma) = Dir(\delta_u|\gamma 1 + \mathbf{dc}_u) \tag{3.7}$$

2. *Given a topic k, its item distribution φ_k and brand distribution ψ_k follow Dirichlet posteriors parameterized by $\phi 1 + \mathbf{c}_k$ and $\alpha 1 + \mathbf{bc}_k$ respectively. Thus, we have:*

$$P(\varphi_k|\mathscr{I},\ \mathscr{Z}, \mathscr{D},\ \phi) = Dir(\varphi_k|\phi 1 + \mathbf{c}_k) \tag{3.8}$$

$$P(\psi_k|\ \mathscr{I},\ \mathscr{Z}, \mathscr{D}, \alpha) = Dir(\psi_k|\alpha 1 + \mathbf{bc}_k) \tag{3.9}$$

3. *Given a brand b, its item distribution follows Dirichlet posterior parameterized by $\beta 1 + \mathbf{c}_b$. Thus, we have*

$$P(\omega_b|\mathscr{I}, \mathscr{B},\ \mathscr{D}, \beta) = Dir(\omega_b|\beta 1 + \mathbf{c}_b) \tag{3.10}$$

Since all the parameters follow Dirichlet posteriors, the expectation of Dirichlet posteriors can be used to estimate them. We skip the details here but interested readers can easily find them in any standard reference on Dirichlet posterior e.g. .

3.4 Experiments

To evaluate BIT against two baselines LDA and sBIT, we first conduct experiments using synthetic adoption data that contains ground truth labels, i.e., item's topic label, adopter's brand-consciousness, brand's topic label, and brand's exclusiveness. We also vary the dataset parameters to study how BIT performs under different data settings.

3.4.1 Experiments on Synthetic Data

Data generation The set of parameters used in synthetic data generation is given in Table 3.3. For simplicity, every brand is assigned to only one topic. Each item is associated with R_{brand} brands, and thus R_{brand} topics. Every user is assigned K favorite topics and $N_{topic} - K$ non-favorite topics where N_{topic} is the total number of topics. $P\%$ ($P > 50$) of adoptions are reserved for items in the user's favorite topics leaving the remaining $100 - P\%$ to those in non-favorite topics. $Q\%$ of users are brand-conscious and they adopt items based on exclusive brands. $X\%$ of brands for each topic are designated as exclusive brands. We also impose the constraint that each brand has at least 10 items. This ensures enough brand-based adoption data for each brand.

Using the parameters as listed in Table 3.3, we generate the synthetic data as follows: 1) For each brand, randomly assign a topic label while ensuring that every

Table 3.3: Parameters for Synthetic Data Generation

Symbols	Description	Value Range (Default Value)
N_{user}	# users	10K
N_{brand}	# brands	100
N_{item}	# items	1K
N_{topic}	# topics	$\{5, 10, 15\}$ (10)
R_{adopt}	# adoptions/user	[50,200] (100)
P	% adoptions in favorite topics	90
Q	% brand lovers	[0, 100] (20)
X	% exclusive brands/topic	10

topic has similar number of brands. 2) For each topic, randomly designate $X\%$ of these brands to be exclusive. 3) For each brand, randomly assign 10 items to ensure that each brand later will have at least 10 items. 4) For each item, randomly assign two brands. 4) For each user, randomly assign two topics as his favorites. 5) Randomly assign $Q\%$ of users to be brand conscious and they will always adopt items of exclusive brands. 6) Every user u is assigned the same number of adoptions R_{adopt} and to generate each adoption of u, first select one of two favorite topics of the user. If he is brand conscious, randomly select an exclusive brand under the topic followed by randomly selecting an item under the exclusive brand. Otherwise (i.e., the user is non brand conscious), randomly select an item under the favorite topic.

Results

Topic-item distribution error. All three models BIT, sBIT and LDA learn the topic assignments of item adoptions. We aim to evaluate the accuracy of models' topic assignments with respect to the ground truth topic assignments. Since all models are instances of unsupervised learning, we will not be able to exactly recover the ground truth topics after learning. We first have to match the learned topics with the ground truth topics and examine how accurate the matching is. For each model, the matching procedure is described below.

- Given a topic z (either ground-truth or learned), denote its item distribution as $I(z)$. For each ground-truth topic k_t, we first determine k_l, its *best matched* learned topic, to be the one whose item distribution is *closest* to that of the true topic k_t. The closeness is measured by Jensen-Shannon (JS) distance. Thus k_l and the corresponding error in recovering ground-truth topic k_t can be defined as following.

  $k_l := \text{argmin}_{z_l} JS\left[I(z_l), I(k_t)\right]$ and $Err(k_t) := JS\left[I(k_l), I(k_t)\right]$

- By taking average of all $Err(k_t)$, we can define error $TopicErr$ in learning

Q	t-test	*sBIT*	*LDA*
0%	*BIT*	2.8E-14*	1.5E-01
25%	*BIT*	6.3E-10*	1.7E-04*
50%	*BIT*	2.0E-09*	1.2E-06*
75%	*BIT*	1.7E-08*	9.2E-08*
100%	*BIT*	6.6E-01	5.7E-12*

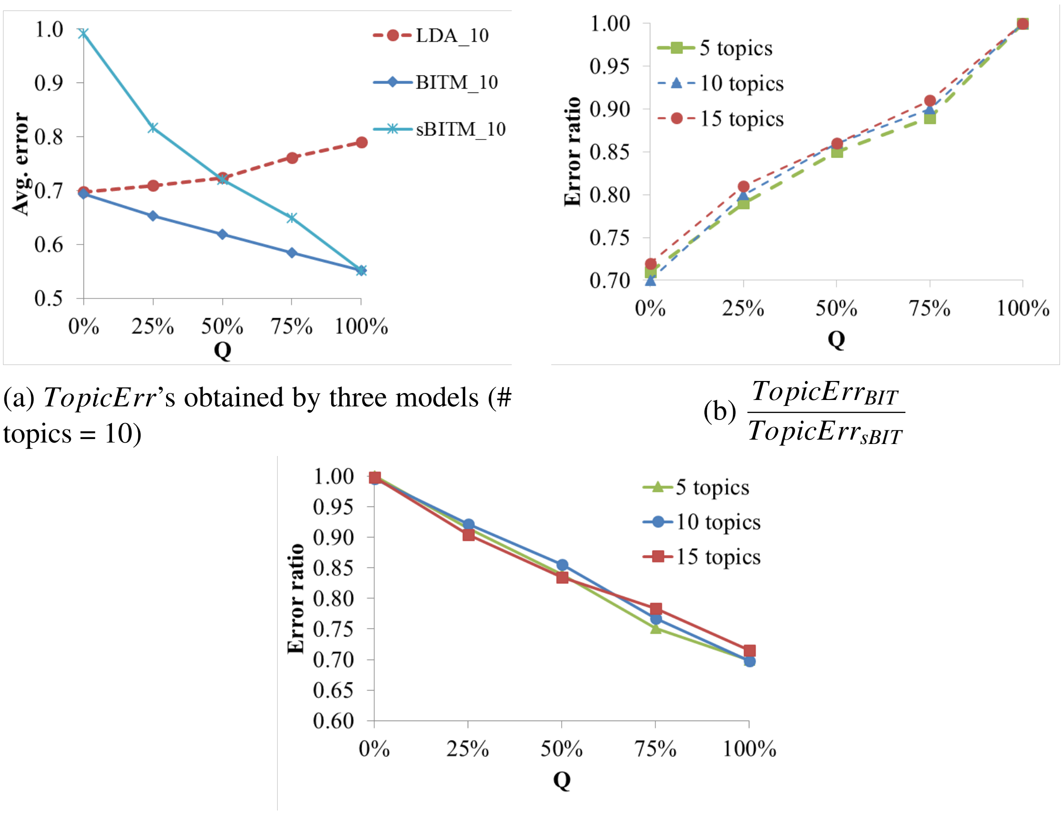
Table 3.4: p values from paired t-tests (2-tail) on errors in learning topics. **Note**: $^*p < 0.01$.

topics for each model as

$$TopicErr := avg_{k_t} Err(k_t) \tag{3.11}$$

Figure 3.2a shows the errors obtained by the three models on learning topic-item distribution where we fix the number of topics as 10 while varying % of brand conscious users Q from 0 to 100. As expected, BIT and LDA produce the same error values when $Q = 0$ whereas BIT and sBIT performs the same when $Q = 100$. As Q increases, both BIT and sBIT show that they can learn the topic labels more accurately when there are more brand conscious users. LDA, on the other hand, generates larger error when Q increases. Finally, the performance of sBIT is worse than LDA if less than 50% of users are brand conscious; which is reasonable since more than 50% of adoption decisions are now topic-based. We also performed paired t-tests (see Table 3.4) to check if these results are statistically significant. It can be seen that BIT improves significantly over sBIT (LDA) when $Q < 100$ ($Q > 0$) respectively.

To verify if the improvement by BIT is consistent we vary the number of topics as 5, 10 and 15 respectively. For all settings, we examine topic-item distribution error ratios between BIT and sBIT (Figure 3.2b); between BIT and LDA (Figure 3.2c). Figure 3.2b shows that BIT outperforms sBIT in learning topic-item distribution when $Q < 100\%$ whereas Figure 3.2c shows that BIT outperforms LDA when $Q > 0$. Moreover, given the same Q, the error ratios for three settings of number of topics are also similar. These results again confirm that BIT is the best



(a) $TopicErr$'s obtained by three models (# topics = 10)

(b) $\frac{TopicErr_{BIT}}{TopicErr_{sBIT}}$

(c) $\frac{TopicErr_{BIT}}{TopicErr_{LDA}}$

Figure 3.2: Topic-item distribution errors by various % of brand conscious users Q

among the three models.

Accuracy of brand conscious user prediction: Every user is assigned to be either brand conscious or not brand conscious. We use U_q to denote the set of brand conscious users in the ground truth data, and U'_q to denote the set of brand conscious users learned (or predicted) using BIT. Ideally, we want $U_q = U'_q$. To measure how accurate are the brand conscious users predicted by BIT, we utilize the *Accuracy* measure as defined:

$$Acc_q(Q) = \frac{|U_q \cap U'_q| + |(U - U_q) \cap (U - U'_q)|}{|U|}$$

Figure 3.3 shows that the accuracy of predicted brand conscious users improves with increasing $Q\%$. Compared with a random 50-50 guess which has a 0.5 accuracy, BIT can predict brand conscious users quite well with mostly 0.8 accuracy when $Q\%$ is larger than 20%.

Topic-brand distribution error: In the synthetic data, each brand is assigned a ground truth topic. Using the topic-item distributions, we determine the best matched ground truth topic k'_l for each learned topic k_l. Let the topic-brand distribution of k_l and k'_l among item adoptions by brand conscious users be denoted by $A(k_l)$ and $A(k'_l)$ respectively. We define the *topic-brand distribution error* (denoted by $BrandErr$) between learned and ground truth topic-brand distributions using Jensen-Shannon divergence measure similar to that for $TopicErr$. As shown in

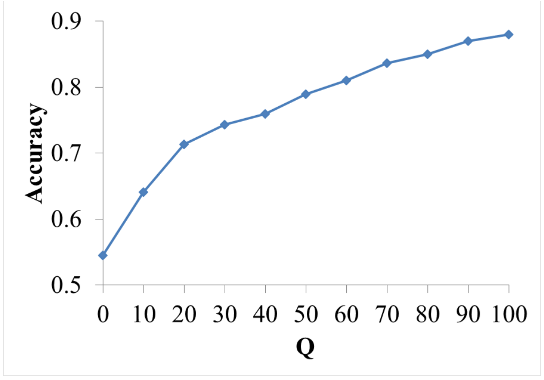


Figure 3.3: Accuracy of BIT in predicting brand conscious users

Figure 3.4, BIT's topic-brand distribution error improves with larger $Q\%$ of brand conscious users and BIT outperforms sBIT when $Q\%$ is less than 100%. LDA is not involved in this evaluation as it does not learn the exclusive brands for each topic.

3.4.2 Experiments on Real Data

We conducted a series of experiments on the BIT model using two real world datasets derived from Foursquare and ACM Digital Library (ACMDL) . We first derive subsets of the datasets using a sampling strategy that trims away users with very few adoptions. The experiments then seek to uncover the hidden topics and brand preferences in item adoptions using BIT. We also compare the topics derived from BIT with those from LDA.

Datasets.

Our Foursquare dataset consists of check-in data generated by Singapore users from October 2012 to April 2013. Each food outlet is an item, each food outlet chain is a brand and each check-in is an item adoption by a user. From the raw dataset, we selected a subset of the data based using top $k = 100$ brands and denote the selected data as **4SQDB**. The selection steps will be elaborated shortly.

For ACMDL, each citing author is a user, and each publication is an item. Every publication belongs to one or more authors who are also treated as brands in our experiments. Each citation of some publication by some (citing) author is an item

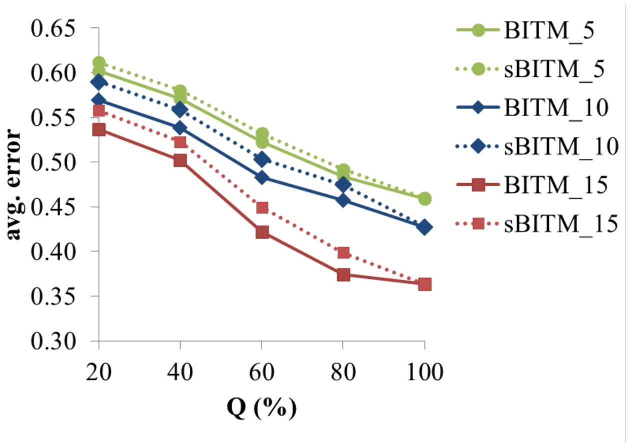


Figure 3.4: Topic-Brand Distribution Error

Table 3.5: Data Statistics

Dataset	# users	# brands	# items	# adoptions
4SQDB	5406	622	2444	64,622
ACMDB	356	7520	3790	16,308

adoption. We used publications in ACMDL from 1998 to 2005 to select a subset using top $k = 10$ authors. The selected data is denoted as **ACMDB**. We explain the steps of choosing for both datasets 4SQDB and ACMDB with the aid of Figure 3.5.

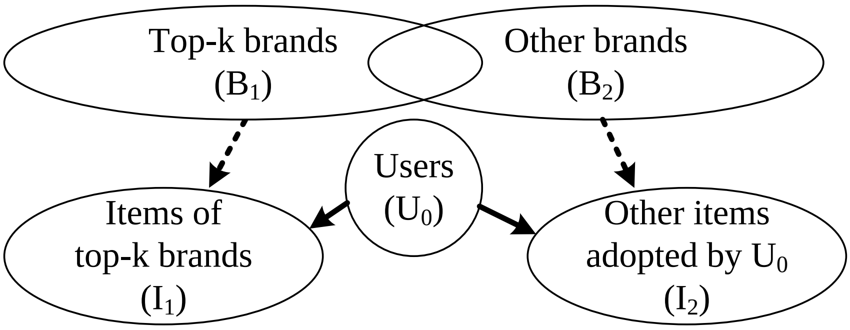


Figure 3.5: Dataset construction (solid line = adoption, dash line = brand relationship)

First, we selected top k brands based on the number of item adoptions of the brands. We denote the set of brands as B_1. Then, we selected all items that belong to B_1 and denote this set of items as I_1. Next, we extracted users that have adopted at least one item in I_1. This set of users is denoted by U_0. Then we extracted all other items adopted by U_0. We denote the new set of items as I_2. We extracted all brands of items in $I_0 = I_1 \cup I_2$ and denote this set as B_0. Finally, we filter away users in U_0 with less than two item adoptions, items in I_0 with less than two adoptions from users in U_0, and brands in B_0 that have no items. We repeat this filtering step until all the remaining users, brands and items satisfy the minimum thresholds. We denote the final sets of users, brands and items as U, B, and I. The statistics of two obtained datasets 4SQDB and ACMDB are shown in Table 3.5.

Prior parameters.

To determine appropriate prior parameters, we performed grid search and chose optimal parameters which maximizes log likelihood function. After grid search, we

Table 3.6: **4SQDB**: JS divergence between topic-item distributions learned by BIT and LDA

	AL1	AL2	AL3	AL4	AL5	AL6	AL7	AL8	AL9	AL10	AL11	AL12
AB1	**0.701**	0.915	0.927	0.852	0.940	0.838	0.859	0.756	0.805	0.778	0.728	0.843
AB2	0.967	**0.530**	0.976	0.971	0.671	0.835	0.689	0.974	0.855	0.971	0.947	0.870
AB3	0.934	0.916	**0.675**	0.891	0.870	0.842	0.770	0.955	0.702	0.902	0.905	0.965
AB4	0.933	0.954	0.855	**0.717**	0.941	0.794	0.806	0.905	0.838	0.733	0.924	0.932
AB5	0.883	0.889	0.931	0.910	**0.580**	0.962	0.858	0.937	0.770	0.893	0.881	0.931
AB6	0.877	0.905	0.910	0.903	0.871	**0.516**	0.906	0.893	0.802	0.820	0.816	0.879
AB7	0.889	0.906	0.870	0.877	0.903	0.816	**0.552**	0.878	0.810	0.853	0.874	0.873
AB8	0.935	0.936	0.917	0.950	0.944	0.831	0.794	**0.546**	0.870	0.890	0.944	0.812
AB9	0.885	0.908	0.909	0.924	0.858	0.937	0.829	0.879	**0.611**	0.871	0.854	0.913
AB10	0.958	0.944	0.968	0.917	0.871	0.924	0.819	0.829	0.908	**0.717**	0.945	0.869
AB11	0.894	0.942	0.705	0.949	0.958	0.867	0.807	0.913	0.819	0.964	**0.675**	0.862
AB12	0.898	0.898	0.890	0.914	0.921	0.838	0.841	0.920	0.829	0.928	0.919	**0.697**

got the following values for priors:

$$\alpha = \beta = 0.1; \quad \gamma = 1; \quad \phi = 0.2; \quad \theta = 50/K;$$

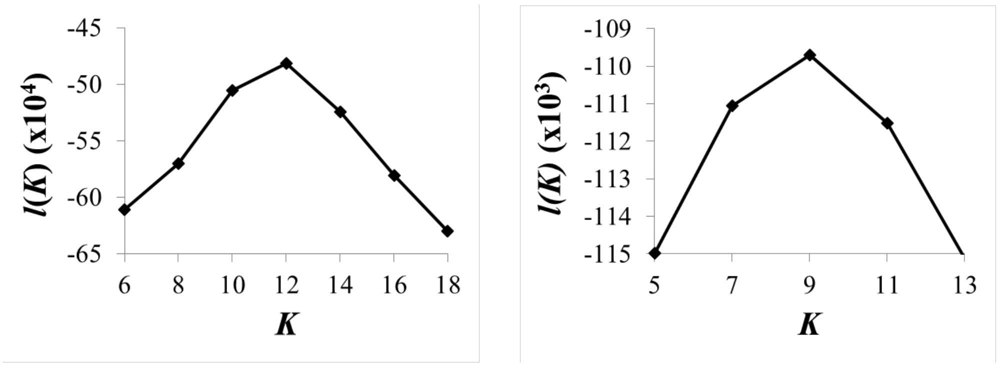
where K is the number of topics.

Result

1. *Topic Analysis*

We first determine the appropriate number of topics for analysing each dataset by running LDA on them. The results in Figure 3.6 show that the log likelihoods reach maximum at 12 and 9 topics for **4SQDB** and **ACMDB** respectively. Thus, in training models, we empirically used 12 and 9 topics for **4SQDB** and **ACMDB** respectively.

4SQDB: We compare the item distributions of topics discovered by BIT and



(a) 4SQDB

(b) ACMDB

Figure 3.6: Log likelihood upon training LDA on two datasets. Here $l(K)$ is the log likelihood w.r.t. the number of topics K.

Table 3.7: **ACMDB**: JS divergence between topic-item distributions learned by BIT and LDA

	TL1	TL2	TL3	TL4	TL5	TL6	TL7	TL8	TL9
TB1	**0.545**	0.937	0.914	0.835	0.914	0.897	0.637	0.899	0.749
TB2	0.918	**0.354**	0.845	0.867	0.832	0.853	0.902	0.871	0.918
TB3	0.820	0.783	**0.673**	0.795	0.829	0.827	0.844	0.831	0.822
TB4	0.862	0.810	0.821	**0.689**	0.804	0.763	0.829	0.810	0.837
TB5	0.852	0.806	0.770	0.843	**0.685**	0.818	0.839	0.829	0.823
TB6	0.828	0.791	0.815	0.796	0.807	**0.758**	0.835	0.798	0.822
TB7	0.813	0.773	0.815	0.805	0.793	0.795	**0.432**	0.824	0.814
TB8	0.846	0.814	0.801	0.780	0.784	0.790	0.851	**0.763**	0.828
TB9	0.867	0.832	**0.696**	0.798	0.787	0.772	0.872	0.776	0.854

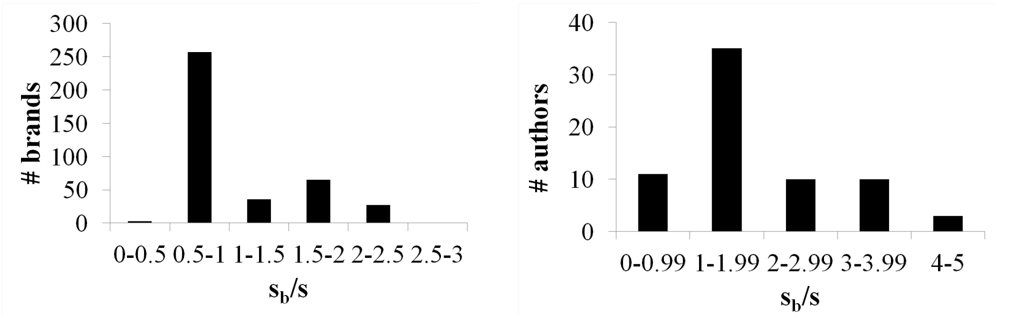
LDA as shown in Table 3.6. AL*n* (AB*n*) represents the n^{th} topic learnt by LDA (BIT). The similarity between item distributions of two topics is given by the Jensen-Shannon (JS) divergence where smaller JS divergence values indicate higher similarity. From Table 3.6, we observed that the two models learned quite similar topics as most values in the diagonals of Table 3.6 are relatively smaller compared to the non-diagonals.

Contrary to our intuition, the learned topics are not about cuisine types (e.g., Chinese food, Indian food) but are clusters of food outlets in 12 different location areas of Singapore as shown in Table 3.8.

ACMDB: We manually determined each topic based on keywords in top-20 titles of that topic. Due to space constraint, the topics discovered and their top-20 paper titles are not provided here but interested readers can find them at extended result [81]. We then compare the topics found by BIT and LDA using JS divergence (Table 3.7). TL*n* (TB*n*) represents the n^{th} topic learnt by BIT (LDA) for ACMDB. In Table 3.7, the columns (rows) show topics learned by LDA (BIT) respectively. Given that smaller JS divergence implies higher similarity, we found that among 9 topics, BIT and LDA agree on 8 topics shown by the bolded diagonal entries of Table 3.7. These topics are Databases and Data Mining (DB+DM), Power Optimization (PO), Software Engineering (SE), World Wide Web (WWW), System, Security, Wireless Network (WN), Computer Architecture (CA). But LDA discovered the topic information retrieval (IR) which BIT did not. Instead, BIT discovered two sub-topics of software engineering: SE_1 (Algorithms and Programming) and SE_2 (Fault Localization).

Table 3.8: Matching learnt topics

Topics (LDA)	Topics (BIT)	Topic Label
4SQDB		
AL1	AB1	Tampines
AL2	AB2	Chua Chu Kang
AL3	AB3	Ang Mo Kio
AL4	AB4	Orchard
AL5	AB5	Pasir Ris
AL6	AB6	Punggol
AL7	AB7	Toa Payoh
AL8	AB8	Hougang
AL9	AB9	Sembawang
AL10	AB10	Jurong
AL11	AB11	Compass Point
AL12	AB12	Bukit Panjang
ACMDB		
TL1	TB1	DB+DM
TL2	TB2	PO
TL3	TB3	SE
TL4	TB4	WWW
TL5	TB5	Systems
TL6	TB6	Security
TL7	TB7	WN
TL8	TB8	CA



(a) 4SQDB

(b) ACMDB

Figure 3.7: Histograms of s_b/s derived from brand conscious users learned by BIT.

2. Brand Preference Analysis

As demonstrated in synthetic experiment, BIT can learn brand preferences of users. It can help to determine whether a user is brand conscious or not. In our experiments, a user is considered as *brand conscious* if at least 80% of his/her adoptions are brand-based. Once the set of brand conscious users is determined, it can be used to identify *exclusive* brands for each topic. This is a major gain provided by BIT as previous models do not help to identify exclusive brands. More specifically, we can identify exclusive brands based on the quantities defined below.

1. $s = \frac{|U_{BC}|}{N_{users}}$ where U_{BC} is the set of all brand conscious users; s can be considered as the *average* ratio of brand conscious users.

2. For each brand b, we define the brand-specific ratio $s_b = \frac{|U_{BC}^b|}{N_{users}^b}$ where U_{BC}^b (N_{users}^b) is respectively the set of *brand conscious* users (the set of all users) who adopted items of brand b.

Note that on estimating s_b, we filtered out brands with $N_{users}^b < 5$ to avoid getting brands with large s_b by pure coincidence. After obtaining these quantities, we compare s_b of each brand with average value s using the ratio s_b/s.

Based on the distributions of the ratio s_b/s shown in Figure 3.7, we propose that exclusive brands (of both 4SQDB and ACMDB) are those for which $s_b/s \geq 2$. This is an appropriate threshold as an exclusive brand should have its s_b much higher than the average value s.

4SQDB: Recall that $N_{users} = 5406$ and BIT learned that $|U_{BC}| = 1319$, thus

$s = 0.24$. There are 29 brands which satisfies $s_b/s \geq 2$. Thus, we can say that BIT learned 29 *exclusive* brands. In Table 3.9a, we show top-10 brands with largest s_b/s as representatives of exclusive brands for 4SQDB. We further checked the reliability of the result by looking at prices of these brands from sg.openrice.com, a popular website for rating food venues in Singapore. The prices are shown in the last column of Table 3.9a. Moreover, on comparing with another 29 less-exclusive brands (those with highest $s_b/s < 2$), the average price of exclusive brands is much higher than that of less-exclusive brands (**20.4** SGD compared with **9.8** SGD) while the standard deviation is comparable (**7.2** compared with **6.0**).

ACMDB: Recall that $N_{users} = 356$ and $|U_{BC}| = 58$, thus $s = 0.16$. Again, we determined exclusive authors as those whose $s_b/s \geq 2$. There are 23 authors satisfying this. Thus, BIT discovered 23 exclusive authors for this ACMDB dataset. Table 3.9b shows top-10 authors with largest s_b/s as representatives of exclusive authors. We further checked the reliability of the result by looking at h-index of these authors provided by Google Scholar. The h-indices are shown in the last column of Table 3.9b. Moreover, on comparing with another 23 *less-exclusive* authors (those with highest $s_b/s < 2$), the average h-index of exclusive authors is much higher than that of less-exclusive authors (**60.5** compared with **34.5**) while the standard deviation is smaller (**14.2** compared with **17.4**).

3.4.3 Summary

Through the above analysis of topics and brand preferences, we demonstrate the usefulness of BIT model. Ideally, these empirical results should be further compared with ground truth topic and brand preference labels. In the absence of ground truth in 4SQDB and ACMDB, we further evaluate the BIT model in item adoption prediction task as described in Section 3.4.4.

Table 3.9: Discovered exclusive brands for two datasets

(a) 4SQDB

Brand	**Area**	s_b/s	**Price (SGD)**
The Halia	AB4	2.53	31-50
Ichiban Sushi	AB7	2.48	21-30
Sushi Tei	AB3	2.39	21-30
Nakhon Kitchen	AB8	2.29	11-20
ThaiExpress	AB12, AB10	2.19	11-20
Pepper Lunch	AB3	2.19	11-20
Pizza Hut	AB2	2.19	11-20
Sakae Sushi	AB8	2.19	11-20
Uncle Leong Seafood	AB5, AB6	2.19	11-20
Astons Specialities	AB9	2.18	11-20
Swensen’s	AB1, AB11	2.18	11-20

(b) ACMDB

Author (i.e. brand)	**Topic**	s_b/s	**h-index**
Giovanni de Micheli	PO	5.00	73
Jon M. Kleinberg	WWW	4.06	69
David Karger	SE_1	4.03	70
Ion Stoica	WWW	3.98	65
Tian Zhang	DB + DM	3.87	59
Leslie Lamport	System	3.85	57
H. T. Kung	WN+ WWW	3.57	55
Jon Louis Bentley	SE_2	3.33	47
M. Frans Kaashoek	CA	3.13	45
John K. Ousterhout	SE_1	3.13	45

3.4.4 Adoption Prediction

We define the item adoption prediction task as follows. For each user u with at least 4 item adoptions, we randomly hide p $(0 < p < 1)$ of these adoptions as the test data. The task is to predict these hidden item adoptions using the remaining $(1-p)$ of adoptions to train a model.

Unlike in the standard recommendation problem where no item is rated again by the same user, the same item can be adopted by the same user in both training and test data. For example, the same paper can be cited by the same authors in multiple papers, and the same outlet can be checked-in multiple times by the same user.

We evaluate the prediction results using *average precision at k* (*AvgPrec@k*) which is defined to be the average of *Prec@k* over all users with adoptions to be predicted. Let I_u^k be the top k predicted adopted items for user u ordered by $p(i_j|u)$, the probability of user i generating the adoption of item i_j. The precision at k for user u, $Prec@k(u)$, is defined as:

$$Prec@k(u) = \frac{|Test_u \cap I_u^k|}{k}$$

where $Test_u$ denotes the set of item adoptions of user u to be predicted.

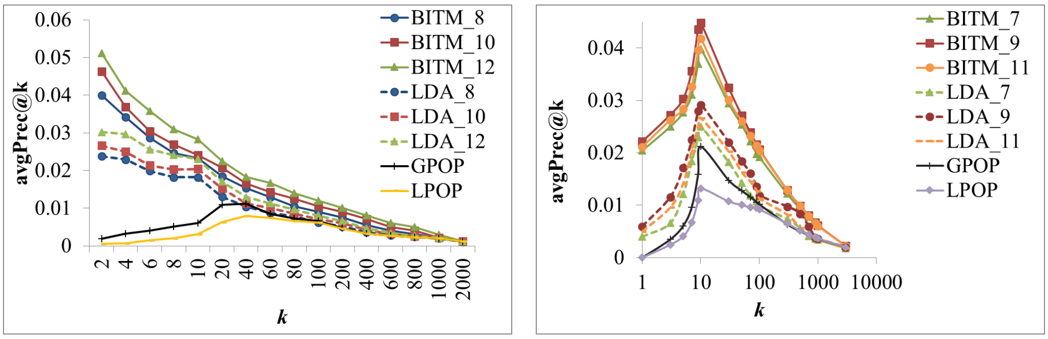
To ensure the results are robust, we conducted 4-fold and 5-fold cross validation of the training and testing data for 4SQDB and ACMDB respectively, and reported the average results. We vary k from 2 to 2000 for 4SQDB, and from 1 to 3000 for ACMDB.

Other than BIT and LDA, we also introduce two other simple baselines, namely:

- *Global Popularity* (**GPOP**): Each item is assigned a global popularity score defined by the number of adoptions it has. Usually, GPOP is not appropriate for prediction task that involves items of very different characteristics. In this experiment, however, the items involved are similar. We therefore include GPOP and also include the local popularity score below.

- *Local Popularity* (**LPOP**): For each user, we assign each item a local popularity score defined by the number of adoptions the user has performed on the item. The items are then ranked by decreasing local popularity score. For each user, his top ranked items are returned as the predicted adoptions.

Note that GPOP returns the same adoption predictions for all users while LPOP returns the frequently adopted items by the target user.



(a) **4SQDB**

(b) **ACMDB**

Figure 3.8: Prediction results w.r.t different number of topics for two datasets. BIT_i (LDA_i) are BIT (LDA) trained with i topics respectively.

Figures 3.8a and 3.8b show the *AvgPrec@k* of prediction results on 4SQDB and ACMDB respectively by varying k and number of topics. The key observations from these figures are that:

- BIT is consistently the best performing model and it is followed by LDA, GPOP and LPOP. We observe this for both datasets for almost all k's and all number of topics.

- For **4SQDB**, the *AvgPrec@k* of BIT and LDA decreases with increasing k. This suggests that the top ranked predicted adoptions by the two models are more accurate than the lower ranked predicted adoptions. For **ACMDB**, we however observe that *AvgPrec@k* increases initially until k reaches about 10. Beyond that, *AvgPrec@k* decreases with larger k. This observation holds for all the models.

- The optimal number of topics for both BIT and LDA for the **4SQDB** dataset is 12 while that for **ACMDB** is 9. This observation is consistent with the numbers of topics determined for the two datasets by likelihood.

To sum up, BIT shows promising prediction results in this experiment and the results are also consistent for both datasets under across different settings.

Limitation of BIT: Though BIT has been demonstrated to have promising performance on moderate size datasets, it is not efficient enough. We profiled the running time of BIT's inference algorithm and identified the following cause. For a given adoption $j = (u, n)$, sampling decision d_j and brand b_j variables separately can lead to the case $d_j = 0$ but a brand is still sampled for b_j. In other words, there is a subtle constraint that whenever the decision variable is 0, the corresponding brand variable must also be 0 (i.e. no brand is sampled). The problem with BIT is that in its process of sampling latent variables, there are times when the constraint is violated. We have observed that this violation slows down the original sampler since BIT needs many iterations to finally overcome this invalid case.

Proposed solution: To enforce the constraint, we propose to *sample jointly* d_j and b_j as a joint variable $y_j = (d_j,\ b_j)$. For a given topic-based adoption j where $d_j = 0$, b_j must also be 0 due to the constraint. Thus, $y_j = (0, 0) = 0$ is the only valid joint variable for topic-based adoptions. When $d_j = 1$, b_j must be some brand b and $y_j = (1, b)$. In short, we define a new kind of latent variable y which receives values in the following set.

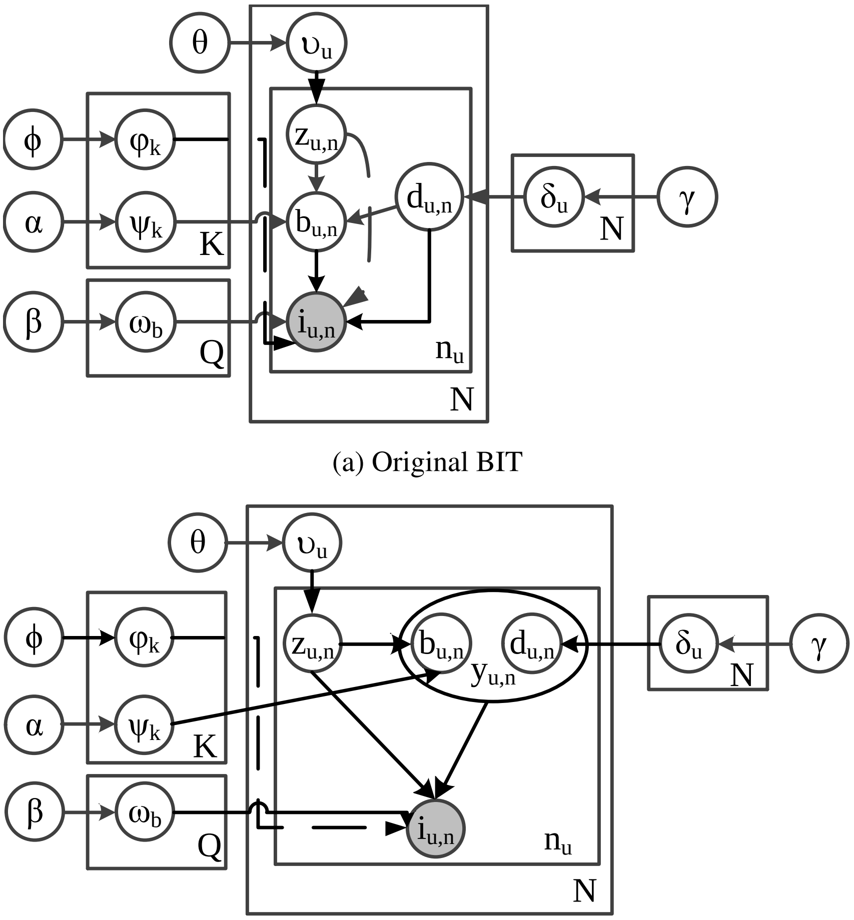
$$S = \{(0,0), (1, b_1),\ \ldots,\ (1, b_Q)\}$$

Given this kind of joint variable, we now can propose in Section 3.5 an enhanced version of BIT, which we call eBIT.

3.5 Enhanced Brand Item Topic (eBIT)

3.5.1 Generative Process of eBIT

The major difference between eBIT (Figure 3.9b) and BIT (Figure 3.9a) is that in eBIT, decision and brand variables are sampled together in a joint variable $y =$



(a) Original BIT

(b) eBIT with joint latent variable

Figure 3.9: Graphical representations of BIT and eBIT

(d, b). It is easy to see that the distribution of joint variables can be derived from the user-decision distribution δ_u and topic-brand distribution ψ_k as follows.

Definition 2. *(Distribution for joint variable) Given user u and topic k, the conditional probability of* $y = (d, b)$ *is then*

$$p(y|u,\ k) := \begin{cases} \delta_{u,0}, & \textit{if } y = (0,0) \\ \delta_{u,1} \times \psi_{k,b}, & \textit{if } y = (1,b) \end{cases} \tag{3.12}$$

Once topic k and joint variable y is known, an item for adoption can be sampled from the following distribution.

$$p(i|k,y) := \begin{cases} p(i|k) = \varphi_{k,i}, & \text{if } y = (0,0) \\ p(i|b) = \omega_{b,i}, & \text{if } y = (1,b) \end{cases} \tag{3.13}$$

Given the generative process of eBIT, we again can use collapsed Gibbs sampling to perform *alternating* inference for the model. This means that in each iteration of Gibbs sampling, we alternatingly sample topics and joint variables. Details of the inference are provided in Section 3.5.2.

3.5.2 Inference of eBIT

Notations

We represent an adoption dataset $\mathscr{D}$ by adoption histories of users i.e. $\mathscr{D} = \{i_u : i_u = (i_{u,1}, \ldots, i_{u,n_u})\}_{u \in U}$. We use the shortened index $j = (u,n)$ to denote the n-th adoption of a specific user u. We also use the symbol ˜ to denote a known value (e.g. $\tilde{k}$ is a known topic). Similar to LDA's inference by Gibbs sampling, the major components of the inference are the counts of latent variables. We summarize all notations for the counts in Table 3.10.

Table 3.10: Notations used in eBIT inference

Notation	**Description**
n_u	Adoption counts of user u.
$\mathbf{tc}_u = (tc_{u,1}, \ldots, tc_{u,K})$ $\mathbf{dc}_u = (dc_{u,0}, dc_{u,1})$ $\mathbf{ic}_k = \left(ic_{k,1},\ \ldots, ic_{k,M}\right)$ $\mathbf{bc}_k = \left(bc_{k,1},\ \ldots, bc_{k,Q}\right)$ $\mathbf{ic}_b = \left(ic_{b,1},\ \ldots, ic_{b,M}\right)$	Topic counts of user u Decision counts of user u Item counts of topic k Brand counts of topic k Item counts of brand b
$i_u = \{i_{u,1},\ \ldots, i_{u,n_u}\}$ $z_u = \{z_{u,1},\ \ldots, z_{u,n_u}\}$ $y_u = \{y_{u,1},\ \ldots,\ y_{u,n_u}\}$	Adoptions by user u. Latent topic assignments. Latent pair assignments.
$\mathscr{I} = \bigcup_u i_u$ $\mathscr{Z} = \bigcup_u z_u$ $\mathscr{Y} = \bigcup_u y_u$	Set of adoptions from all users. Set of all latent topics. Set of all latent pairs.

Inference equations

Latent variable inference: Consider a j-th adoption where u adopts item $\tilde{i}$ i.e. $i_j = \tilde{i}$. We now show the equations for sampling the *latent topic* z_j given the remaining topics $\mathscr{Z}^{-j}$ and $\mathscr{Y}$. We then show the equations for sampling the *latent joint variable* y_j given remaining joint variables $\mathscr{Y}^{-j}$ and $\mathscr{Z}$. Due to space constraint, the derivation of equations is provided in Appendix A.2.

- (Sampling topic) We can sample topic z_j from the following distribution.

$$P(z_j = k|\mathscr{Z}^{-j},\, y_j) \propto \begin{cases} w_0(k), \text{ if } y_j = 0 \\ w_1(k), \text{ if } y_j = (1, \tilde{b}) \end{cases} \tag{3.14}$$

where

$$w_0(k) := (tc_{u,k} + \theta - 1) \times \frac{ic_{k,\tilde{i}} + \phi - 1}{\sum_i ic_{k,i} + M\phi - 1} \tag{3.15}$$

and

$$w_1(k) := (tc_{u,k} + \theta - 1) \times \frac{bc_{k,\tilde{b}} + \alpha - 1}{\sum_b bc_{k,b} + Q\alpha - 1} \tag{3.16}$$

- (Sampling joint variable) Given that $z_j = k$, we can now sample the joint variable $y_j = (d_j,\, b_j)$ from the following distribution.

$$P(y_j = \tilde{y}|\mathscr{Y}^{-j},\, \mathscr{Z}) \propto \begin{cases} w(0,0), \text{ if } \tilde{y} = 0 \\ w(1,\, \tilde{b}), \text{ if } \tilde{y} = (1,\, \tilde{b}) \end{cases} \tag{3.17}$$

where

$$w(0,0) := (dc_{u,0} + \gamma - 1) \times \frac{ic_{k,\,\tilde{i}} + \phi - 1}{\sum_i ic_{k,\,i} + M\phi - 1} \tag{3.18}$$

and

$$w(1,\, \tilde{b}) := (dc_{u,1} + \gamma - 1) \times \frac{bc_{k,\,\tilde{b}} + \alpha - 1}{\sum_b bc_{k,b} + Q\alpha - 1} \times \frac{ic_{\tilde{b},\,\tilde{i}} + \beta - 1}{\sum_i ic_{\tilde{b},i} + M\beta - 1} \tag{3.19}$$

Distribution inference: Once the Gibbs sampling converges, we can infer the five posterior distributions by combining each final count with the corresponding prior. This is possible due to the conjugacy between Dirichlet and multinomial distributions [48]. For example, the posterior probability that u selects topic $\tilde{k}$ is:

$$p(\tilde{k}|u) = \frac{tc_{u,\tilde{k}} + \theta}{\sum_k tc_{u,k} + K\theta} \tag{3.20}$$

3.5.3 Likelihood of eBIT

We denote parameters estimated by eBIT, i.e. the five matrices of distributions, as follows.

1. User-topic matrix of size $N \times K$: $\Theta^T = \left(\vartheta_u^T\right)_{u \in U}$
2. User-decision matrix of size $N \times 2$: $\Delta = \left(\delta_u^T\right)_{u \in U}$
3. Topic-brand matrix of size $K \times Q$: $\Psi^T = \left(\psi_z^T\right)_{z \in Z}$
4. Topic-item matrix of size $K \times M$: $\Phi = \left(\varphi_z^T\right)_{z \in Z}$
5. Brand-item matrix of size $Q \times M$: $\Omega = \left(\omega_b^T\right)_{b \in B}$

Let $\Pi = \{\Theta,\ \Delta,\ \Psi,\ \Phi,\ \Omega\}$ represent all the parameters.

Given parameters Π, the likelihood of the whole adoption dataset $\mathscr{D}$ is just the product of individual likelihoods:

$$P(\mathscr{D}|\Pi) = \prod_u P(i_u|\Pi) \tag{3.21}$$

where the likelihood of each adoption history i_u can be estimated by the following equation:

$$\begin{aligned} P(i_u|\Pi) &= \prod_{i \in I} [p(i|\Pi)]^{n_{u,i}} \quad (n_{u,i}\text{: frequency } u \text{ adopted } i) \\ &= \prod_{i \in I} \left[\delta_{u,0} \times \sum_z \vartheta_{u,z} \varphi_{z,i} + \delta_{u,1} \times \sum_z \sum_b \vartheta_{u,z} \psi_{z,b} \omega_{b,i} \right]^{n_{u,i}} \end{aligned} \tag{3.22}$$

Thus, the log likelihood function will be

$$\begin{aligned}\mathscr{L}(\Pi) &= \log P(\mathscr{D}|\Pi) \\ &= \sum_{\substack{u\in U\\ i\in I}} n_{u,i} \log \left[\delta_{u,0} \times \sum_z \vartheta_{u,z}\varphi_{z,i} + \delta_{u,1} \times \sum_z \sum_b \vartheta_{u,z}\psi_{z,b}\omega_{b,i} \right]\end{aligned} \tag{3.23}$$

3.5.4 Non-identifiability of eBIT

Rewriting the log likelihood (LL) of Equation 3.23 in the following matrix form reveals that eBIT can face non-identifiability issue.

$$\begin{aligned}\mathscr{L}(\Pi) &= \log P(\mathscr{D}|\Pi) \\ &= \sum_{\substack{u\in U\\ i\in I}} n_{u,i} \log \left(\delta_{u,0} \times [\Theta^T \Phi]_{u,i} + \delta_{u,1} \times [\Theta^T \Psi^T \Omega]_{u,i} \right)\end{aligned} \tag{3.24}$$

Indeed, given parameters Π_1, we can design another parameters Π_2 with the same LL using the idea of orthogonal transformation as follows.

Let $\mathscr{O}$ be an orthogonal matrix of size $K \times K$ satisfying $\mathscr{O}^T \mathscr{O} = \mathbf{I}$. As the following assignment keeps the matrix products in the LL unchanged, it also preserves the LL of eBIT.

$$\Theta_2 = \mathscr{O}\Theta_1,\ \Phi_2 = \mathscr{O}\Phi_1,\ \Psi_2 = \Psi_1 \mathscr{O}^T,\ \Omega_2 = \Omega_1 \tag{3.25}$$

Indeed, we have the following equalities:

$$\Theta_2^T \Phi_2 = \Theta_1^T \mathscr{O}^T \mathscr{O} \Phi_1 = \Theta_1^T \Phi_1$$

and

$$\Theta_2^T {\Psi_2}^T \Omega_2 = \Theta_1^T \mathscr{O}^T \mathscr{O} \Psi_1^T \Omega_1 = \Theta_1^T \Psi_1^T \Omega_1$$

This analysis shows one of the possibilities which cause non-identifiability to BIT models. In fact, when we performed experiments on synthetic data for DeBIT the first few times, we observe that this issue usually happens when the generated topics have similar brand or item distributions. Thus, for synthetic data, we tried resolving it by making the topics distinguishable, e.g., generating topics with small brand overlapping. Later experiments show that this way of data generation actually reduces significantly non-identifiability issue. For real data, we initialize inference with several different parameters and check if any two initial parameters share the same likelihood. If that happens, we analyze the topic-item and topic-brand distributions returned by them and choose the initial parameters which provide more reasonable distributions.

3.6 Distributed Enhanced Brand Item Topic (DeBIT)

We build our Distributed enhanced Brand Item Topic (DeBIT) based on the Stale Synchronous Parallel (SSP) framework and the so-called Petuum system built on top of the framework [25, 49]. Thus, we first give a brief review of the framework in Section 3.6.1 and then describe how we employ the framework's ideas and Petuum system to build DeBIT in Section 3.6.2.

3.6.1 Stale Synchronous Parallel Framework

Essentials of the framework include the following.

- A so-called *parameter server* provides shared interface for reading/writing sufficient statistics of a model (i.e. parameters of the model). After a certain number of computations, workers can write their part of updated sufficient statistics to the parameter server. The write operation is then synced among workers after each iteration.

- Workers can read older, stale versions of the sufficient statistics from a local cache, instead of waiting for their updated values from the server. This significantly increases the proportion of time workers spend on useful computations, as opposed to waiting for synchronization.

- The SSP framework ensures the correctness of the model learning by limiting the maximum age of the stale values by a configuration argument called *staleness*, denoted as s. This argument basically controls the consistency level for parameters by forcing that the slowest and fastest workers must be $\leq s$ iterations apart.

We employ the parameter server architecture because it offers several benefits. Firstly, it allows us to concentrate on implementing the essential parts of our algorithm without worrying about low-level communication among machines. It also reduces overhead of communication among machines by supporting various relaxed consistency levels. We thus can choose the optimal one by varying the staleness s.

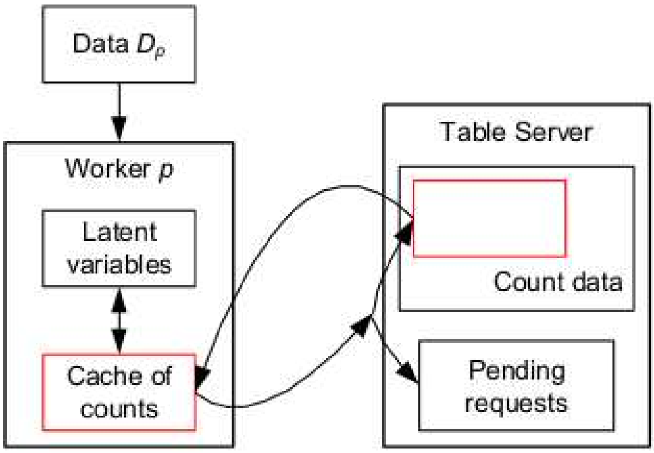


Figure 3.10: High-level structure and mechanism of DeBIT, where each worker is in charge of updating latent variables and counts for a partition D_p of adoption data.

3.6.2 Structure and Mechanism of DeBIT

The structure has two main components (i) workers and (ii) table servers (Figure 3.10).

- Workers: Each worker is a thread whose main task is performing Gibbs sampling on one partition of adoption data to iteratively update corresponding latent variables and counts. At the end of each iteration, workers commit their count updates to table servers. For each count variable, the Petuum system will then aggregate its relevant updates into the count to get new count for the next iteration.

- Table servers: The servers are responsible for storing and serving *global* count tables to workers. The count tables are exactly what the Gibbs sampler needs i.e. the co-occurence counts of pairs *user-topic*, *user-decision*, *topic-item*, *brand-item* and *topic-brand*. For simplicity, we re-use notations from Table 3.10 for the count tables.

Given P workers, we first partition the set of users into P disjoint subsets $\{U_p\}_{p=1}^{P}$ each with N/P users. Each worker p will then be assigned adoption data $\mathscr{D}_p$ such that all adoptions of one user go to the same worker. Worker p will then perform a local Gibbs sampler to learn *local* latent variables for the adoption data. Our assignment ensures that workers do not share their local latent variables. Thus, we do not need global tables for latent variables. This can reduce both storage and time consumption as the global tables can be huge for large-scale data. Given this structure and division of data, we repeat the following process until convergence (i.e. change in log likelihood is smaller than a certain threshold).

- For each worker p in parallel, perform Gibbs sampling and then commit updates as follows.

  Gibbs sampling: local latent variables are sampled by replacing current *global* counts into sampling equations (3.14) - (3.19). In the process, relevant counts are also "updated". Specifically, when an old value of a latent variable is replaced by a new value, two update operations are recorded (but not committed yet)

  1. minus 1 for counts of the old value (see Algo. 2)
  2. plus 1 for counts of the new value (see Algo. 2)

  Committing updates: once all updates are ready, p will inform table servers and other workers and commit the updates to the servers.

- At table servers, updates from all workers will be aggregated and added to current counts to give new counts for next iteration.

When it needs counts from a table, worker p first checks its cache to see if the staleness of cached counts are still within the staleness bound s (i.e. $\leq s$ iterations apart from the fastest worker). If that is the case, it will read the stale counts without waiting for getting them from the table servers. Only when the staleness exceeds the bound s, it will send a read request to the table servers. As mentioned in SSP framework in Section 3.6.1, this mechanism helps to reduce time spent on communication so that workers can spend more time on computational works. Details on how the back-end Petuum system enforce this mechanism of bounded staleness can be found in [49].

3.6.3 Implementation of DeBIT

Each worker performs the following tasks

- Initializing latent variables and corresponding counts for adoptions of assigned users.
- Repeatedly running Gibbs samplers to update the latent variables and the counts until convergence.
- Inferring five distributions from final counts and priors (see "Distribution inference" in Section 3.5.2).

In this implementation, we also let workers perform burn-in to get better guesses for variables. Algorithm 1 and Algorithm 2 provide the pseudocode for the working process of each worker and the Gibbs sampler respectively.

3.7 Experiments on Synthetic Data

We first perform experiments on synthetic data to compare DeBIT, eBIT and BIT in the following aspects.

Algorithm 1 Working process of each worker in DeBIT

```
procedure run($\mathscr{D}_p$, $latents$, $counts$, $thres$, $length$)
    initialize($latents$, $counts$, $U_p$)
    burnIn($length$)
    repeat
        for each $u \in U_p$ do
            gibbSampler($latents$, $counts$, $u$)
        end for
        commitUpdatesToServers()
    until convergence
    ▷ Centralize all parameters at one worker and output them
    if isFirstWorker() then
        $distributions \leftarrow$ inferFrom($counts$, $priors$)
        save($distributions$)
    end if
end procedure
```

Table 3.11: Parameters used for synthetic data generation

Symbols	Description	Value range	Default value
N	# users	$1000, 3000, 5000$	1000
K	# topics	$5, 10, 15$	10
Q	# brands	$20, 40, 60, 80, 100$	60
M	# items	$200, 400, 600, 800, 1000$	600
A	# adoptions	$\{1, 3, 5\} \times 10^6$	1×10^6

(a) Data dimensions

Symbols	Description	Value range (%)	Default value (%)
p_{bca}	Fraction of BCUs	$0, 10, 20, 30, 40, 50$	10
p_{ex}	Fraction of exclusive brands	$0, 5, 10, 15, 20$	5

(b) Fractions

- Efficiency: in terms of time or number of iterations to convergence.

- Accuracy: in terms of recovering ground-truth (i) user-topic, topic-item, brand-item and user-decision distributions; (ii) brand-conscious users (abbreviated as BCUs); and (iii) exclusive brands.

For these purposes, we generate adoption datasets based on the principle that brand conscious users tend to adopt from exclusive brands. Parameters for data generation are summarized in Table 3.11.

Algorithm 2 DeBIT's Gibbs sampler for a user u

```
procedure gibbsSampler($latents$, $counts$, $u$)
    for each adoption $j$ of $u$ do
        incTopicCounts($u, z_j, d_j, b_j, i_j$, $-1$)
        Sample new topic $z_j^n$ by Eqns. (3.14) – (3.16)
        incTopicCounts($u, z_j^n, d_j, b_j, i_j$, 1)

        incJointCounts($y_j, u, i_j, z_j^n$, $-1$)
        Sample new joint variable $y_j^n$ by Eqns. (3.17) – (3.19)
        incJointCounts($y_j^n, u, i_j, z_j^n$, 1)
    end for
end procedure

procedure incTopicCounts($u, z, d_j, b_j, i_j$, $val$)
    $tc[u,z] \leftarrow tc[u,z] + val$
    if $d_j == 0$ then    $ic[z,i_j] \leftarrow ic[z,i_j] + val$
    else   $bc[z,b_j] \leftarrow bc[z,b_j] + val$
    end if
end procedure

procedure incJointCounts($y_j, u, i_j, z_j^n$, $val$)
    ▷ Proceed similarly as incTopicCounts, details in [82]
end procedure
```

3.7.1 Data Generation

To simulate real world situations, we generate a dataset embedded with *brand-conscious* users and *exclusive* brands. Moreover, we allow topics to have overlapping brands and each brand to have one flagship (i.e. the most popular) item. Given the numbers of users, topics, brands, items and adoptions denoted as N, K, Q, M and A respectively, we obtain such a dataset using the following steps.

1. **(User generation)** We first label randomly a proportion p_{bca} of N users as *brand-conscious*. We then assign all adoptions of a (non) brand-conscious user to be (non) brand-based respectively. We then generate topic preference for each user u by assigning randomly to u a favorite topic, leaving the $(K-1)$ remaining topics as non-favorite.

2. **(Brand assignment)** We label randomly a proportion p_{ex} of Q brands as *exclusive*. We then assign Q brands uniformly to topics such that any two topics

have $\leq 10\%$ of their brands overlapping. Specifically, each topic is assigned $q = \lfloor Q/(K - 0.1) \rfloor$ relevant brands, of which $p_{ex}q$ are exclusive.

3. **(Item assignment)** We assign $m = M/Q$ items to each brand b and let one of them to be the flagship item.

   Each topic is then assigned qm relevant items from its q relevant brands. For each topic k, we then assign $1/5$ of its items to be popular, such items occuppy $4/5$ of topic-based adoptions under k. Thus, ground-truth *topic-item* distributions are:

$$\varphi_{k,i} = \begin{cases} 4/(qm), \text{ if } i \text{ is a popular item of } k \\ 1/(4qm), \text{ if } i \text{ is a normal item of } k \\ 0, \text{ otherwise} \end{cases} \tag{3.26}$$

4. **(Adoption generation)** For each user u, we generate each of A/N adoptions of u by two steps.

   - (Topic sampling) we sample topic k from u's topic distribution

$$\vartheta_{u,k} = \begin{cases} 0.7, \text{ if } k \text{ is } u\text{'s favorite topic} \\ 0.3/(K-1), \text{ otherwise} \end{cases} \tag{3.27}$$

   - (Item sampling) If u is brand conscious, we sample an item i directly from topic k's item distribution (Eqn. (3.26)). Otherwise, we sample a brand b from k's brand distribution

$$\psi_{k,b} = \begin{cases} 1/(p_{ex}q), \text{ if } b \text{ is exclusive} \\ 0, \text{ otherwise} \end{cases} \tag{3.28}$$

and then sample i from brand b's item distribution

$$\omega_{b,i} = \begin{cases} 0.8, \text{ if } i \text{ is } b\text{'s flagship item} \\ 0.2/(m-1), \text{ if } i \text{ is normal} \\ 0, \text{ otherwise} \end{cases} \tag{3.29}$$

3.7.2 Efficiency and Scalability Evaluations

We set up these experiments as follows.

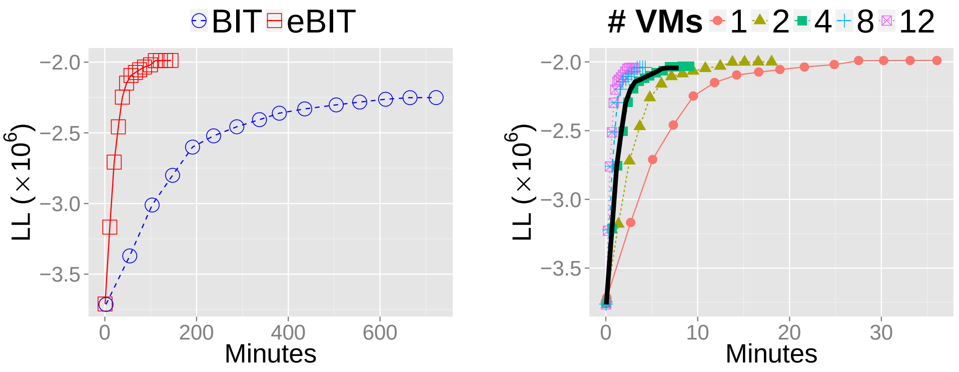
- **Computing cluster:** Multi-core servers connected by 1Gbps Ethernet, running VMware. We use three virtual machines (VMs) per physical machine. Each VM is configured with 4 cores (Intel Xeon E5, 2.9GHz each) and 70GB of RAM, running on top of Centos Linux 6.
- **Staleness:** s is varied from 0 to 5. The purpose is to see how staleness affects/helps with efficiency and accuracy.

In our parameter tuning experiments, we determined that staleness $s = 2$ and $s = 3$ yielded the least convergence times. Actually, the convergence speed of DeBIT with 12 VMs and $s = 0$ (the black line in Figure 3.11b) is only equivalent to that of DeBIT with 4VMs and $s = 2$. When s is larger than 3, the accuracy starts to degrade as the log likelihood at convergence gets smaller. Thus, we fix $s = 2$ in all efficiency and scalability evaluations.

Efficiency results: We compare efficiency of the models on the same dataset D_1 with $N = 1000$ users, $K = 10$ topics, $Q = 60$ brands, $M = 600$ items and $A = 1$ million adoptions. As shown in Figures 3.11a and 3.11b, we can see that

- eBIT converges much faster (about 5 times) than BIT.
- DeBIT's convergence speed increases with more VMs.

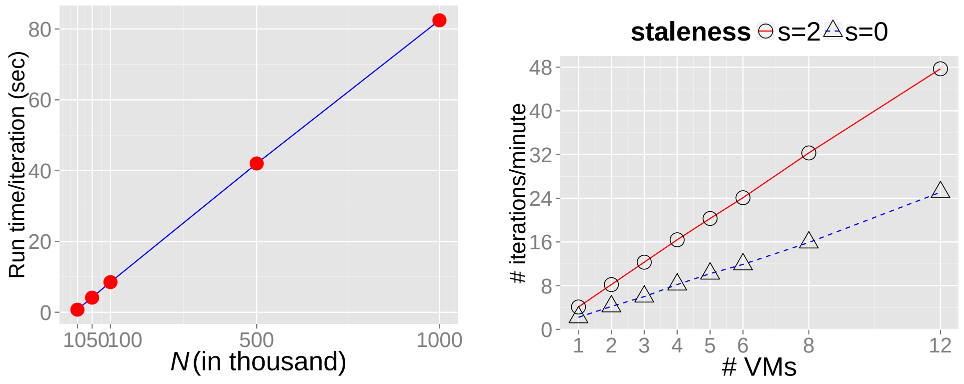
Scalability results: To evaluate scalability of the models in terms of number of users N, we fix other parameters (i.e. $K = 10$ topics, $Q = 60$ brands, $M = 600$



(a) **eBIT** vs. **BIT**

(b) **DeBIT**: various # VMs ($s = 2$; black line: $s = 0$, 12 VMs)

Figure 3.11: Efficiency of models



(a) Scaling in number of users (12 VMs and $s = 2$)

(b) Iteration throughput scaling

Figure 3.12: DeBIT's scalability

items) and vary N from 1,000 to 1,000,000 (A is varied from 50,000 to 50 million). We observe that

- DeBIT scales linearly with N (as shown in Figure 3.12a).

- There is a linear speed-up in number of iterations per minutes as a function of number of VMs (as shown in Figure 3.12b). Moreover, we can see that staleness $s = 0$ gives inferior performance than $s = 2$. The reason is that $s = 0$ causes much larger communication overhead.

- For the largest dataset of size 1GB (1 million users and 50 million adoptions), it takes about 4.5 hours for DeBIT to finish one round of training (150 iterations: 50 for burn-in and 100 for actual training). This is highly efficient if we compare against BIT, which can only deal with less than one million adoptions using the same computing configuration.

3.7.3 Accuracy Evaluations

In this section, we compare the accuracy of models eBIT, DeBIT and BIT. The computing cluster for DeBIT is the same as in Section 3.7.2. We elaborate on how to evaluate accuracy of the models in terms of (i) recovering ground-truth distributions; and (ii) discovering brand-conscious users and exclusive brands.

Recovering ground-truth distributions

First, we compare the models' accuracy on the same dataset. Secondly, we examine how accuracy of each model changes with the ratio p_{bca} of brand-conscious adopters in data.

Datasets: For these purposes, we generate datasets with different p_{bca} values in $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ and use the default settings of other parameters.

Baseline: We will use topic model LDA as baseline to evaluate the accuracy of our models in learning topic-item and user-topic distributions as they are similar to topic-word and document-topic distributions in LDA.

Metrics: Jensen-Shannon (JS) distance from learned distributions to their *corresponding* ground truth ones. JS distance is a popular similarity measure for probability distributions (see [34, 75]). It is suitable for our accuracy evaluations since it is bounded in $[0, 1]$ (see [34]).

Since our models learn topics in an unsupervised manner, their learned topics are different from ground-truth topics (usually by a permutation). Thus, we first need to point out how to match ground truth topics with those learned by each model. Once the matching is done, it is then possible to compare distributions involving topics — namely topic-item, user-topic and topic-brand. Given a model μ, we now elaborate on its corresponding matching and error computations for the three distributions.

Definition 3 (Matching topics)**.** *Given the k-th ground-truth topic represented by* *item distribution φ_k, we define $\pi_\mu(k)$, its* best match *learned by model μ, as the learned topic whose item distribution is closest to φ_k.*

$$\pi_\mu(k) := \underset{j}{\operatorname{argmin}} JS[\widehat{\varphi}_j; \varphi_k] \tag{3.30}$$

where $\widehat{\varphi}_j$ denotes the j-th item distribution learned by μ.

We then compute model error in recovering the mentioned distributions as follows.

Definition 4 (Topic-item distributions error)**.**

$$TopicErr_\mu := \frac{\sum_{k=1}^{K} JS[\widehat{\varphi}_{\pi_\mu(k)}, \varphi_k]}{K}$$

The gain of μ over baseline LDA is then

$$gain_\mu^{topic} := \frac{TopicErr_{lda} - TopicErr_\mu}{TopicErr_{lda}} \tag{3.31}$$

Definition 5 (Topic-brand distributions error)**.**

$$Err_\mu^{tb} := \frac{\sum_{k=1}^{K} JS[\widehat{\psi}_{\pi_\mu(k)}; \psi_k]}{K} \tag{3.32}$$

After reordering each learned user-topic distribution $\widehat{\vartheta}_u$ by the matching (i.e. replace its k-th element by $\pi_\mu(k)$-th element), we get the error of μ in recovering user-topic distributions (a.k.a topic preference).

Definition 6 (User-topic distributions error)**.**

$$PrefErr_\mu := \frac{\sum_{u=1}^{N} JS\left(\widehat{\vartheta}_u, \vartheta_u\right)}{N}$$

The gain of μ over baseline LDA is then

$$gain_{\mu}^{pref} := \frac{PrefErr_{lda} - PrefErr_{\mu}}{PrefErr_{lda}} \tag{3.33}$$

Finally, for user-decision distributions, we do not need to do any matching. Thus, we can directly compute the JS distance from each user's learned distribution to his ground-truth distribution as follows.

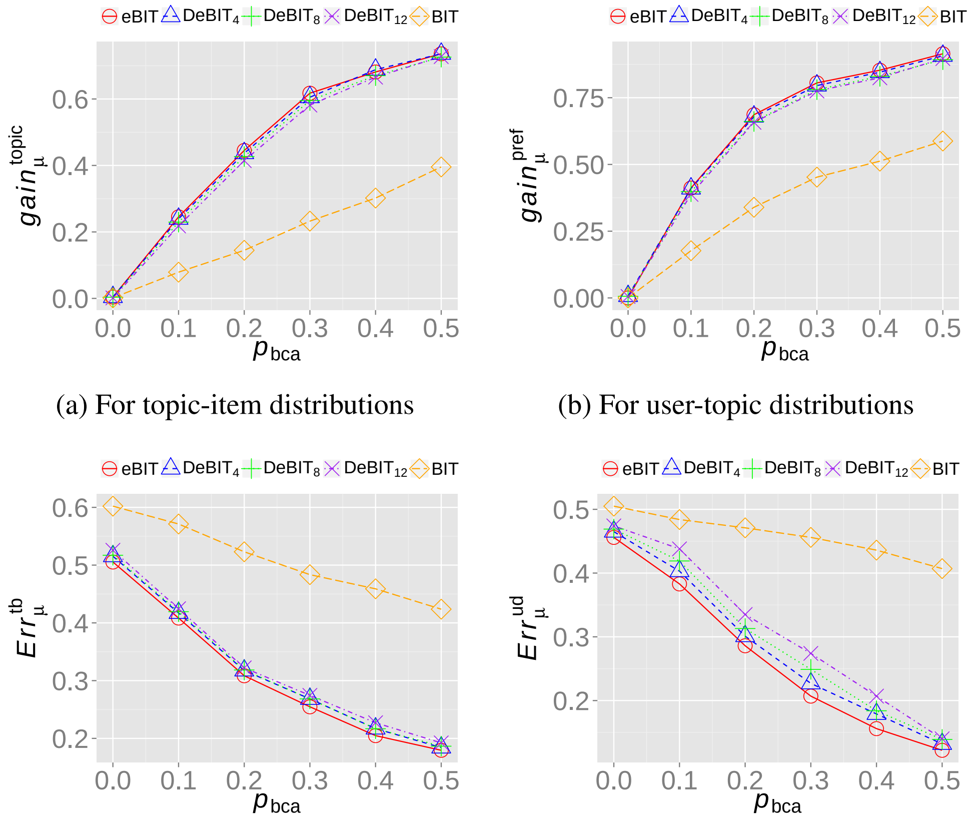
$$Err_{\mu}^{ud} = \frac{\sum_{u=1}^{N} JS[\widehat{\delta}_{u}^{\mu}, \delta_{u}]}{N} \tag{3.34}$$

In summary, we will use gains over baseline LDA $gain_{\mu}^{topic}$, $gain_{\mu}^{pref}$ and errors $Err_{\mu}^{tb}, Err_{\mu}^{ud}$ for evaluating model performance in recovering ground-truth topic-item, user-topic, topic-brand and user-decision distributions respectively. As for brand-item distributions, the evaluation is very similar and thus skipped for brevity.

Results: The performance of the three models in recovering ground truth topic-item, user-topic, topic-brand and user-decision distributions is given in Figure 3.13. From the figure, we can observe the following.

- The models DeBIT, eBIT, and BIT outperform LDA in learning topics and users' topic preference when there are brand-conscious adopters in data. Moreover, the larger the fraction p_{bca} of these adopters is, the larger is the performance difference.

- More importantly, the performance of eBIT is superior than that of BIT. As shown in Figures 3.13a and 3.13b, while the gains $gain_{eBIT}^{topic}$, $gain_{eBIT}^{pref}$ increase in a super-linear manner, the gains $gain_{BIT}^{topic}$, $gain_{BIT}^{pref}$ only increase in a linear manner. Similarly errors of eBIT decreases much faster than those of BIT (see Figures 3.13c and 3.13d). This superior performance of eBIT can be explained by the usage of joint variable in the model, which allows eBIT to successfully eliminate invalid cases (i.e., the topic-based adoptions incorrectly associated with brands) while BIT has difficulty in doing so.



(a) For topic-item distributions

(b) For user-topic distributions

(c) For topic-brand distributions

(d) For user-decision dists.

Figure 3.13: Accuracy of models in recovering ground-truth distributions; DeBIT is run on 4, 8 and 12 VMs.

- eBIT and its distributed version DeBIT share similar accuracy performance as expected.
- In learning topic-brand and user-decision distributions (Figures 3.13c and 3.13d), both eBIT and DeBIT outperform BIT. Moreover, the errors of both models are small, which means that they can learn the two distributions with high accuracy.

Learning brand-conscious users (BCUs) and exclusive brands

Given that eBIT and DeBIT can estimate the topic-brand and user-decision distributions with high accuracy, we now use the distributions for learning brand-conscious users (BCUs) and exclusive brands. These tasks can be considered as two binary classification tasks. The first is to classify users as brand conscious (positive) vs.

non-brand conscious (negative) and the second is to classify brands as exclusive (positive) vs. non-exclusive (negative).

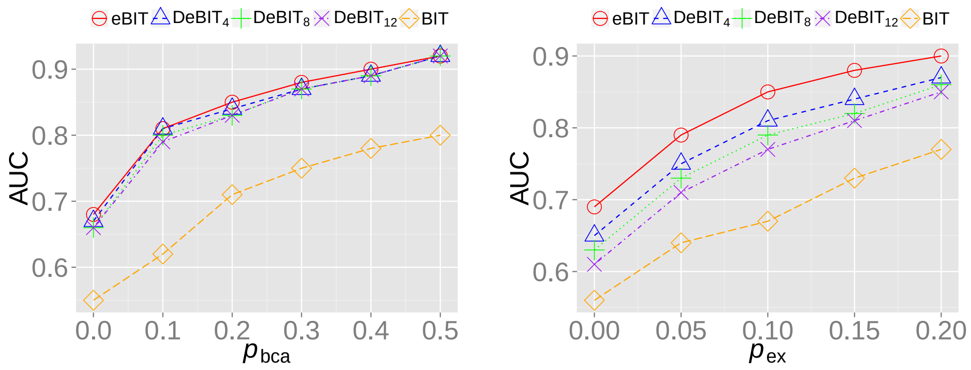
Datasets: To evaluate the performance of the models in the first task, we re-use datasets generated with different ratios of brand-conscious adopters/users p_{bca} in $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. Similarly, for the second task, we generate datasets with different ratios of exclusive brands by varying p_{ex} in $\{0, 0.05, 0.1, 0.15, 0.2\}$.

Baseline: We use BIT model as the common baseline for both tasks as BIT is currently the only model capable of learning the two types of latent variables.

Metrics: Both classification tasks are imbalanced when $p_{bca} \leq 0.2$ or $p_{ex} \leq 0.1$. Thus, we employ Area Under ROC curve (AUC) metrics as our performance measure since the metrics can handle class imbalance issue (see [39, 69]). Specifically, given a dataset $\mathscr{D}$, we evaluate the performance of each model μ as follows.

- (User classification) For each user u, the probability $\delta^{\mu}_{u,1}$ that u makes brand-based adoptions (learned by training μ on $\mathscr{D}$) is used as the level of brand consciousness of u. We then rank users by the probability in decreasing order, construct the corresponding ROC curve by varying the classification threshold and compute the AUC for μ.

- (Brand classification) For each topic k and brand b, the probability $\psi^{\mu}_{k,b}$ (learned by training μ on $\mathscr{D}$) is used as the score for exclusiveness of b under topic k.



(a) Learning BCUs

(b) Learning exclusive brands

Figure 3.14: Performance of models in learning brand-conscious users and exclusive brands

We then rank brands descendingly by the score and compute the corresponding AUC for μ as described above.

Results: From Figure 3.14, we can observe the following.

- *Learning brand-conscious users*. While the baseline BIT does not perform very well when $p_{bca} < 0.2$ (i.e. less than 20% of users are brand-conscious), eBIT and DeBIT perform fairly well even with p_{bca} as low as 10%. Moreover, they start to show good performance with $AUC > 0.8$ when at least 20% of users are brand-conscious.

- *Learning exclusive brands*. For this task, eBIT and DeBIT also outperform the baseline BIT. Moreover, eBIT (DeBIT) starts showing good performance when at least 10% (15% respectively) of brands are exclusive.

To sum up, eBIT and DeBIT outperform baselines LDA and/or the original BIT. Moreover, they can detect brand-conscious adopters (exclusive brands) with high precision even when such adopters (brands) occupy just a small portion of the corresponding population.

3.8 Experiments on Real Data

The experiments aim to evaluate performance of DeBIT in the following tasks.

- Uncovering the *hidden topics* behind item adoptions as well as *topic preference* of each user

- Discovering *exclusiveness of brands* under each topic

For the first two tasks, we train the models on full datasets and we focus on empirical analysis of the learnt results. For the third task, we hide 20% of latest adoption data and train the models on 80% of data. We then use the learned models to make predictions on the hidden data. Details on how to conduct experiment on adoption prediction will be provided later.

3.8.1 Datasets

We conduct experiments on two real datasets: (i) checkin dataset extracted from Foursquare and (ii) citation dataset extracted from the Digital Library of ACM (see [3]). We elaborate on how to collect and preprocess each dataset in following sections. The detailed statistics of both datasets are provided in Table 3.12.

Foursquare checkin

The Foursquare dataset consists of check-in data by Singapore users at food venues in Singapore collected from June 2011 to October 2015. For this dataset, an item is a *food* venue, a brand is a chain of food venue(s), and an item adoption is a check-in by a Foursquare Singapore user.

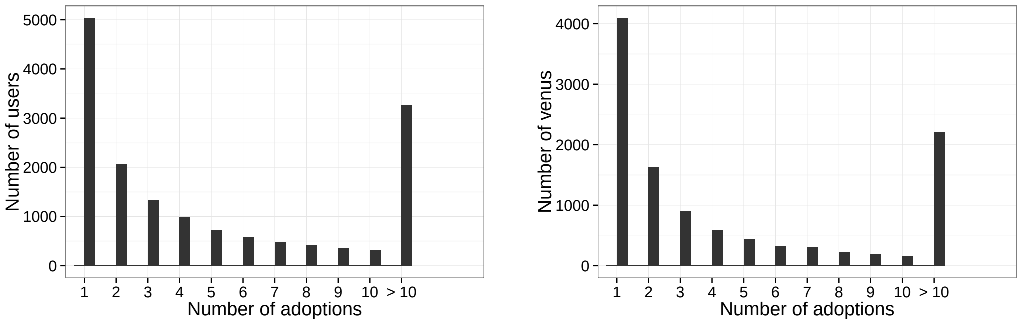
Data collection: At the time of starting this experiment, we did not have a list of Singapore Foursquare users before hand, so we could not collect Foursquare data directly from Foursquare's API. Instead, we collected Foursquare check-ins embedded in tweets published by a set of Twitter users whose profile location is Singapore or geotagged as Singapore. The tweets are collected in real time using the Twitter public stream API. For each collected check-in, we then extracted its related information (Foursquare user profile and venue information) and stored in this dataset. Totally, we collected about 2 million check-ins.

Data preprocessing: We used the categorization from Foursquare to extract check-ins to *food* venues. Among $\approx 330K$ such check-ins, there are approximately $130K$ check-ins to food venues in Singapore. The remaining are check-ins to venues outside Singapore, e.g. when a user travels abroad.

Both distributions of adoption over users and venues are heavy tailed with lots of users and venues having only one or two adoptions (see Figure 3.15). For users

Table 3.12: Statistics of the datasets

Dataset	# users	# items	# brands	# adoptions
4SQDB	$10,301$	$6,748$	$6,237$	$122,628$
ACMDB	$364,259$	$712,926$	$740,890$	$3,178,062$



(a) User adoption counts

(b) Venue adoption counts

Figure 3.15: Heavy-tail adoption distributions in Foursquare data

and venues with a single adoption, it is hard to learn any knowledge as they have too few data, thus we filtered them out. By repeatedly filtering out these users and venues, we finally obtained a dataset where each user and each venue has *at least two adoptions*. We name the final dataset 4SQDB.

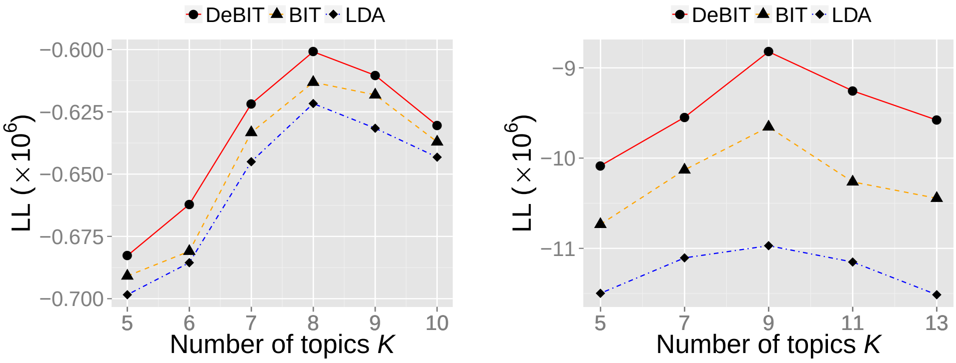
ACMDL citation

The citation dataset is extracted from meta data of publications in ACMDL from 1998 to 2005. For this dataset, an item is a paper, a "brand" is a *cited* author, a user is a the *first* author of a paper who cites other papers and an adoption is a citation.

Data collection and preprocessing: We parse the meta data of publications in ACMDL to extract citation data and authorship information (i.e., who is author of which paper). The authorship information is necessary as it tells us which item (i.e., paper) belongs to which brands (i.e., authors). We then filter out users with less than 10 adoptions. After this, we are left with approximately 360K users. We then extract all adoptions of the users to form our final dataset, denoted as ACMDB.

3.8.2 Models and Hyper-parameters

In these experiments, we compare the performance of DeBIT against those of BIT and LDA. We leave out eBIT as its performance is equivalent to DeBIT. We perform grid search and chose optimal settings for hyper parameters which maximize log likelihood function. The grid search shows that for both datasets, we can use the same $\alpha = \gamma = \varphi = 0.01$, $\theta = 0.1$ as optimal. However, optimal β is 0.01 for 4SQDB and 0.1 for ACMDB. A possible explanation is that the brand-item distribution in the former is sparser than the latter.



(a) 4SQDB

(b) ACMDB

Figure 3.16: Log likelihood of models

3.8.3 Topic Analysis

Since the number of topics K is not known before hand, we train BIT, DeBIT (1 VM, 4 threads) and LDA using different K values. We then observe that for both datasets, our models and LDA agree on the optimal number of topics. For 4SQDB, they all obtain maximum likelihood at 8 topics (Figure 3.16a) while for ACMDB they all reach maximum likelihood at 9 topics (Figure 3.16b).

4SQDB

For each model, we manually labeled each of eight topics based on the top 10 venues (with highest probabilities in the topic's venue distribution). We found that the models also agree on the topics themselves, which are *Breakfast* and seven popular types of cuisine in Singapore: $\{American, Chinese, Indian, Italian, Japanese\}$ *cuisines*, *BBQ* and *Seafood*. More details on the eight topics can be found in Table A.3 of Appendix.

ACMDB

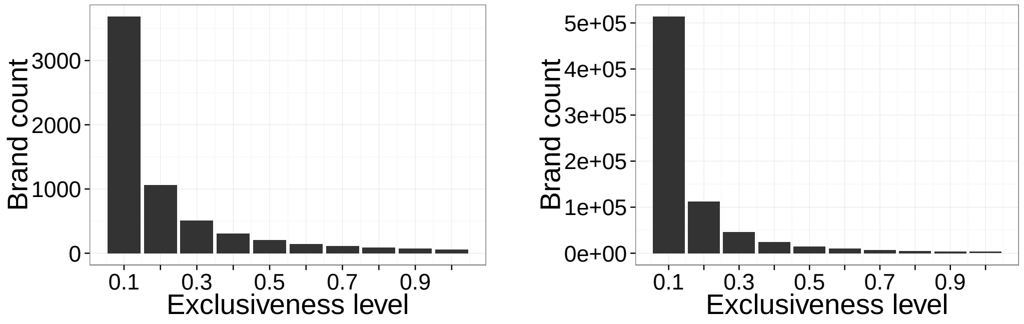
Although all models share the same optimal number of topics for ACMDB, there are some minor differences between topics learned by the models after manual comparison. In fact, all models agree on 8 topics, which are *Databases and Data Mining* (DB+DM), *Power Optimization* (PO), *World Wide Web* (WWW), *System*, *Security*, *Wireless/Sensor Network* (WSN), *Distributed Systems* (DS) and a sub-topic of Software Engineering, namely SE_1 - *Programming*. LDA learned Information Retrieval as the 9th topic. BIT and DeBIT discovered another sub-topic of Software Engineering, namely SE_2 - *Fault Localization* as the 9th topic. More details on the learned topics can be found in Table A.4 of Appendix.

3.8.4 Analysis on Brand Exclusiveness

As we already know, the probabilities from each topic's brand distribution learned by DeBIT can be used to measure the level of exclusiveness of brands with respect to the topic. We measure the overall exclusiveness level of a given brand b by taking the maximum probability as follows.

$$ex(b) := \max_k \psi_{k,\, b} \tag{3.35}$$

For both datasets, the histograms of learned exclusiveness levels show a heavy tail pattern (see Figure 3.17). Specifically, in both datasets, while most of brands ($\approx$



(a) 4SQDB

(b) ACMDB

Figure 3.17: Heavy-tail distributions of brand exclusiveness

80%) have low exclusiveness (≤ 0.2), less than 3% of them have high exclusiveness (≥ 0.8). This is a reasonable distribution for brand exclusiveness.

Brand exclusiveness in 4SQDB: We next validate our exclusiveness results by computing the correlations between overall brand exclusiveness and: (i) *average price* of brand obtained from HungryGoWhere.com website [57], (ii) *checkin count*, and (iii) *user count* available in 4SQDB. These correlations are shown in Table 3.13 as cor_{price}, $cor_{checkin}$, cor_{user} respectively. From Table 3.13, we can see that our exclusiveness has a strong positive correlation with average price of brand and slightly negative correlations with checkin count and user count. This matches our expectation that exclusive restaurants should have higher prices and fewer adopters.

Table 3.13: *4SQDB* — Correlations between exclusiveness learned by DeBIT and different empirical measures

topic	cor_{price}	$cor_{checkin}$	cor_{user}
American	0.737	-0.358	-0.352
BBQ	0.756	-0.348	-0.292
Breakfast	0.815	-0.338	-0.16
Chinese	0.914	-0.045	-0.148
Indian	0.849	-0.184	-0.206
Italian	0.936	0.037	0.024
Japanese	0.801	0.06	0.036
Seafood	0.902	-0.29	-0.106
Average	0.839	-0.183	-0.138

Table 3.14: *ACMDB* — Correlations between exclusiveness learned by DeBIT and citation count

Topic	*cor*	Topic	*cor*	Topic	*cor*
DB+DM	-0.15	SE_2	-0.2	DS	-0.11
PO	-0.33	WWW	-0.4	System	-0.35
SE_1	-0.25	WSN	-0.1	Security	-0.37

Brand exclusiveness in ACMDB: For this dataset, we validate our obtained exclusiveness by computing its correlation with an empirical measure available in ACMDB: citation count of each "brand" (i.e. cited author). We simply denote the correlation as *cor*. Again we found a negative correlation with an average -0.25

(see Table 3.14) which can confirm our expectation that exclusive author has fewer adopters.

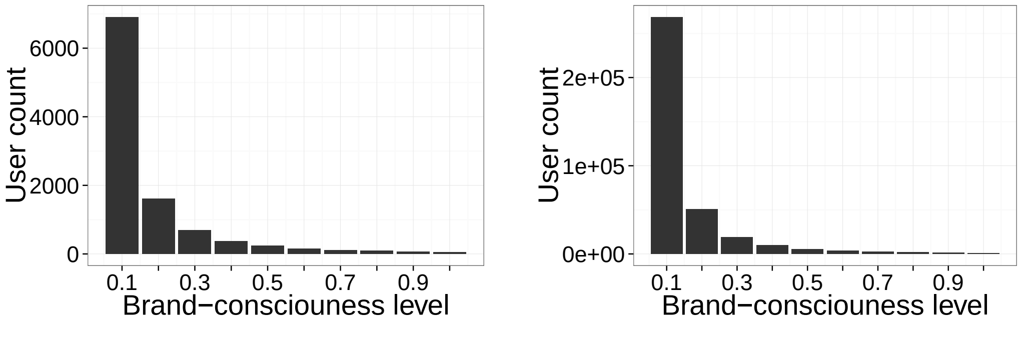
3.8.5 Analysis on User Brand-consciousness

We use the probabilities from user-decision distributions learned by DeBIT as a measure for brand-consciousness level of users. We plot the histograms of the brand-consciousness level for both datasets and observe a heavy tail pattern (see Figure 3.18). For both datasets, while more than 80% of users have low brand-consciousness (≤ 0.2), less than 2% of them have high brand-consciousness (≥ 0.8). Specifically, only 2% (1.4%) of users in 4SQDB (ACMDB respectively) have such high level of brand-consciousness. This result makes sense and matches our intuition that user brand-consciousness should have a heavy tail distribution.

3.8.6 Adoption Prediction

Finally, we compare the prediction power of BIT, DeBIT and LDA. Specifically, we conduct *top-k prediction* as follows.

Training and test sets: For each dataset, we sort the adoptions of each user chronologically. We hide the latest 20% for prediction test and remaining 80% for training. We use 8 topics (9 topics) for all models to make prediction on the test sets of 4SQDB (ACMDB) respectively as these numbers of topics give the maximum likelihood.



(a) 4SQDB

(b) ACMDB

Figure 3.18: Heavy-tail distributions of user brand-consciousness

Adoption probabilities by the models: For a user u, we find his top-k prediction under each model by first estimating the probability that u adopts an item i under the model. We then rank items by their adoption probabilities and predict the top k items with highest probabilities as the items to be adopted by u.

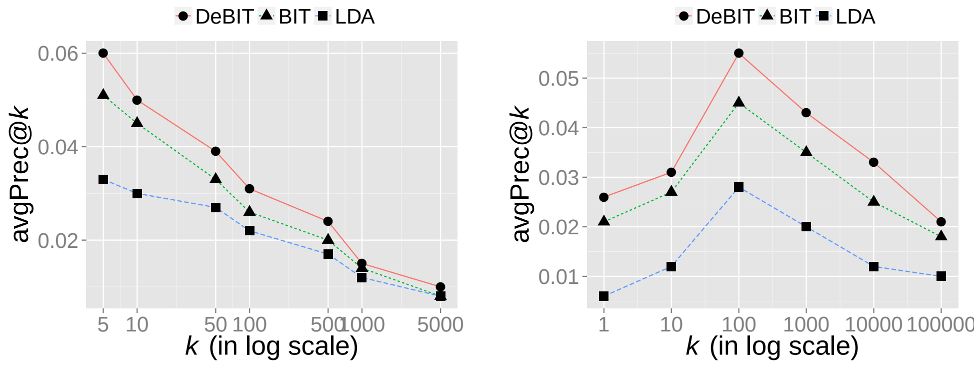
Probability estimated by LDA — under LDA, the only way that u adopts i is through some topic. Thus, by denoting γ_k and θ_u as topic k's item distribution and user u's topic distribution learned by LDA respectively, we obtain the probability that u adopts i as

$$p_{lda}(i|u) = \sum_{z=1}^{K} p(i|z) \cdot p(z|u) = \sum_{z=1}^{K} \gamma_{z,i} \cdot \theta_{u,z} \tag{3.36}$$

Probability estimated by BIT or DeBIT — for these models, u can adopt i either by a topic-based adoption (with probability $\delta_{u,0}$) or a brand-based adoption (with probability $\delta_{u,1}$). Thus, we can use the same Eqn. (3.37) for both models, with implicit understanding that when we switch to DeBIT, the distributions should be replaced accordingly.

$$p_{bit}(i|u) = \widehat{\delta}_{u,0} \times \sum_{z} \widehat{\vartheta}_{u,z} \cdot \widehat{\varphi}_{z,i} + \widehat{\delta}_{u,1} \times \sum_{z} \sum_{b} \widehat{\vartheta}_{u,z} \cdot \widehat{\psi}_{z,b} \cdot \widehat{\omega}_{b,i} \tag{3.37}$$

Adoption prediction: We now make top-k prediction using adoption probabil-



(a) 4SQDB

(b) ACMDB

Figure 3.19: Precision of models in item adoption prediction. All models' predictions on 4SQDB (ACMDB) are made using 8 topics (9 topics respectively).

ities from Equations (3.36) and (3.37) for ranking items. Note that for both datasets 4SQDB and ACMDB, re-adoption can happen, thus we do not exclude previously adopted items when we estimate the probabilities. We then compute *precision@k* for this user (Eqn. 3.38) and we use average of the precisions (*avgPrec@k*) over all users as the metrics for prediction power of the model.

$$precision@k(u) = \frac{|pred_k(u) \cap test(u)|}{k} \tag{3.38}$$

where $pred_k(u)$ and $test(u)$ are respectively the predicted items and the items actually adopted by u in test set.

Results: From Figures 3.19a and 3.19b, we can see that both BIT and DeBIT outperform LDA in the prediction task on both datasets. Moreover, DeBIT gives the best performance. For 4SQDB, all models achieve highest average precision at top-5 items and the average precision descreases with k. This suggests that for Foursquare adoptions, high rank items give better prediction than low rank items. For ACMDB, we however observe that the precision initially increases until $k = 100$ items and decreases as k becomes larger than 100. To sum up, DeBIT shows promising prediction results and the results are consistent for both datasets.

3.9 Discussion

Mining fine-grained yet important knowledge such as *brand exclusiveness* and *user brand consciousness* from adoption data has important applications in marketing and recommendation. In this chapter, we first propose a generative model, namely BIT, which incorporates the two novel concepts and learns new latent variables associated with the two concepts. Inference of the model involves learning different kinds of latent variables and relevant distributions which requires a very large number of variables and parameters for large-scale datasets. Thus, we later tackle the scalability issue by developing DeBIT, a $\underline{d}$istributed and $\underline{e}$nhanced version of BIT.

Our comprehensive experiments on synthetic datasets demonstrate that DeBIT outperforms baselines (LDA and/or BIT) in learning model parameters. It also performs well in discovering brand-conscious users as well as exclusive brands injected in the synthetic datasets, even when these only occupy a small proportion of the user and brand populations. DeBIT is efficient and scalable to large-scale datasets. The scale-up is ideal as a linear function of (i) number of participating workers, and (ii) number of users in data. For a large-scale synthetic dataset of 50 millions of adoptions, while it took eBIT days to finish learning, DeBIT only needed a few hours.

We also investigate the performance of DeBIT on two real-world adoption datasets extracted from Foursquare and ACM Digital Library. Our experiments show that DeBIT not only can discover underlying topics but also *brand-conscious users* and *exclusive brands* from adoption data without requiring further prior information. Our empirical analysis shows that the results on brand-conscious users and exclusive brands are reasonable and interpretable. Specifically, on both datasets, the distributions of these brand-conscious users and exclusive brands are heavy tailed. Moreover, on Foursquare check-in data, the brand exclusiveness learned by our models has strong *positive* correlation with price while on ACM citation data, it has *negative* correlation with citation count. Finally, we demonstrate that DeBIT outperforms baseline LDA in adoption prediction on both datasets.

Although its workers perform local Gibbs sampling based on *inexact* global counts, DeBIT still can perform equivalently well with the single-machine algorithm, which has access to *exact* counts. Thus, a theoretical justification for this phenomenon offers an interesting future reseach question.

Chapter 4

Jointly Modeling Brand and Social Effects in Adoption

4.1 Introduction

Our behaviors of adopting items are determined by several personal factors (e.g., interest, budget constraint, and brand preference); social factors (e.g., friends adopting the same items); item factors (e.g., item features and brand) and other external marketing event factors. Modeling how these factors interact with one another as item adoptions occur is an important research problem as these factors affect the performance of item search and recommendation applications. Ideally, we would like to consider all these factors in a single model. In this work, we address this goal by focusing on modeling item adoptions that can be attributed to brand among other factors.

The presence of brand leads to new challenges in modeling the user-item adoptions. Firstly, every user may have her own brand preferences (or awareness) and users decide items to be adopted with or without considering brand. A user may adopt items based on brand preferences only, on topical interest only, or a mixture of both. When a user does not depend on brand, her brand preferences may become unimportant. On the other extreme, another user may adopt items completely based on brand making the topical interests less important.

The second challenge is brought about by the co-mingling between brand and social connections in adoption decisions. Just as a user's interest topics could be influenced by the interest topics of socially connected users, the user's brand preferences could also be influenced by those of socially connected users. Modeling the brand preferences and brand based adoption decisions in the context of social network is therefore essential.

4.1.1 Research Objectives and Contributions

In summary, brand is very relevant and important to recommender systems but it has not yet been carefully studied in the recommender system research community. Moreover, it is essential for our research to incorporate social network into brand-aware recommender systems. In this paper, we therefore seek to develop a new matrix factorization based recommendation method that considers brand-related factors and social factors influencing user-item ratings. The contributions of this work can be summarized as follows:

- We propose a novel matrix factorization-based recommender system method, called **Social Brand-Item-Topic Model (SocBIT)** that incorporates both brand factors and social homophily. Other than modeling each user and item topic factors, SocBIT assigns each user and item a set of brand factors and learns the brand-consciousness level of users. SocBIT further models social homophily by facilitating socially connected users to share similarity in topic and brand factors.

- We develop a gradient-descent inference for learning the parameters of SocBIT model. As the inference does not force non-negative constraints on topic factors, which makes it hard to interpret such learnt factors, we propose a non-negative version for SocBIT, called **SocBIT**$^+$.

- Our experiments on synthetic data show that our models not only perform better than state-of-the-art methods in recovering ground-truth *topic factors* of users and items but also recover well *brand factors* with small recovery errors. We also demonstrate that SocBIT$^+$ performs well in brand-conscious users detection.

- More importantly our experiments on two real-world datasets from Foursquare and ACM Digital Library (ACMDL) show that both SocBIT and SocBIT$^+$ improve significantly user-item adoption prediction accuracy over state-of-the-art models. The improvement is at least 30% on Foursquare data and 20% on ACMDL data. While SocBIT$^+$ is demonstrated to perform just slightly worse than SocBIT, the former returns non-negative factors, which can easily be interpreted as topic and brand preferences of users.

- Finally, we provide empirical findings which can answer a few research questions, namely: (a) what is the proportion of brand conscious users in the studied user population? (b) how are brand conscious users different from other users? (c) how are the brands preferred by brand conscious users different from those by other users?

The rest of this chapter is organized as follows. In Section 4.2, we review (i) *social* recommendation with a highlight on SoRec model, an inspiration to our SocBIT; and (ii) *brand-based* recommendation with more evidence on the importance of brands in item adoption and recommendation. Next, we provide the formulation and inference of SocBIT and SocBIT$^+$ in Section 4.3. We then show our evaluation experiments on synthetic and real-world datasets in Sections 4.4 and 4.5. Finally, we provide conclusion and discussions on future work in Section 4.6.

4.2 Related Work

Social Recommendation. Inspired by the idea that users' ratings of items may be influenced by users' friends, MF approach has been extended to consider social connections among users [9, 60, 85, 86, 87, 88, 89, 90, 91]. These connections may be friendships, trusts, follow links or others. By incorporating the observed social connection into MF, it has been shown that user latent factors, item latent factors and social influence can be jointly learned. This approach also yields higher recommendation accuracy. A representative of these works is a probabilistic MF model called SoRec [85] which jointly factorizes the rating matrix **R** and user-user social weight matrix **W** into user and item latent factors as

$$\mathbf{W} \approx \Theta_U^T \mathbf{Z} \quad \text{and} \quad \mathbf{R} \approx \Theta_U^T \Theta_I \tag{4.1}$$

where, in addition to the user and item latent factor matrices Θ_U and Θ_I, **Z** is another user latent factor matrix for generating the social weight matrix. However, introducing a second user latent factor matrix **Z** reduces the interpretability of SoRec model. To address the interpretability issue, two other variants of SoRec model were proposed in [86] and [89] respectively. The first, called *Recommendation by Social Trust Ensemble* (RSTE), assumes that neighbors of a user directly influence his *ratings* instead of his latent factors. Thus, the proposed factorization for RSTE is

$$\mathbf{R} \approx \mathbf{W}^T \Theta_U^T \Theta_I \tag{4.2}$$

The second variant, called *Recommendation with Social Regularization*, employs latent factors of a user's neighbors to regularize the latent factors of the user himself. Similarly, authors in [60] also proposed to learn latent factors of a user as a weighted average of his neighbor factors. On the whole, all these models extend the traditional MF approach by incorporating the social network information. Although none of them consider brand factors, SoRec and RSTE are state-of-the-art models and provide important ideas for our approach. Thus, we will later compare our models against the two models in our experiment evaluation.

Brand-based Recommendation. According to Belén del Río et al. [10], brands are important in item adoption and recommendation since they provide the following functions: (i) guarantee, (ii) personal identification, (iii) social identification, and (iv) status.

The *guarantee* function refers to the ability of brands to provide quality assurance, meeting consumer expectations and reducing perceived risks, especially when a consumer has to choose an item in a unfamiliar topic or under uncertainty [6, 35, 37, 38, 123]. The *personal identification* function refers to consumers identifying themselves with some brands. The greater the consistency between the brand image and the consumer's self-image, the larger is her preference toward the brand and the more likely she adopts items from the brand [10, 44, 52]. The *social identification* function refers to the brand's ability to communicate its consumers' desire to be integrated or differentiated from those she interacts with. This comes from the Optimal Distinctiveness theory on the concurrent needs for differentiation and assimilation of consumers [15, 79]. Finally, the *status* function refers to the admiration and prestige a consumer may enjoy if she uses items from a brand [47, 118, 124].

Although brand has such important functions in user-item adoption, brand-based recommendation receives much less attention compared with previous approaches. There are very few works focusing on modeling brand effect on ratings and item adoptions [56, 125, 138]. In [138], the authors studied the correlation between the brands "liked" by a social media user and his items purchased to make recommendations of items of new brands. The work focuses on user brand preference but overlooks the social network information. Meanwhile, the work in [56] proposed a two-module recommendation system consisting of the modules for product profiling and user profiling respectively. The former profiles products based on a given product-brand taxonomy. The latter profiles users to find "brand-sensitive" users, defined in the work as users who are only interested in particular brands. Based on the product and user profiles, the system then calculates the so-called "Recently Repurchase Tendency" scores and makes recommendation based on the scores.

4.3 Proposed Concepts and Models

Before we describe our proposed models Social Brand Item Topic (SocBIT) and its non-negative version SocBIT$^+$, we first define the observed rating and social network data to be modeled. We then describe two empirical analysis on real-world data motivating the assumptions of our models.

In this paper, we use the following standard representation for both synthetic and real-world datasets. Given a set of users $U = \{u_1, \ldots, u_N\}$ and a set of items $I = \{i_1, \ldots, i_M\}$, the ratings of the users on the items are represented by a $N \times M$ user-item matrix $\mathbf{R} = (\{r_{u,i}\})$. A rating $r_{u,i}$ is undefined when u has not rated i. Otherwise, $r_{u,i}$ can be any real numbers in $[0,1]$ after normalization. The users U are connected by a (un)directed social network $G = (U, \mathbf{W})$ where $\mathbf{W}$ is the $N \times N$ matrix of non-negative weights. A positive weight $w_{u,v}$ represents the strength of social (tie) influence between user u and user v, while a zero weight $w_{u,v}$ represents no connection. We require the weights to be in $[0,1]$. We represent the brand-create-item relationship by a $Q \times M$ matrix $\mathbf{B}$ with binary values. $b_{i,j} = 1$ when item i belongs to brand j, and 0 otherwise. In short, the standard representation of a dataset in this work is $\mathscr{D} = (\mathbf{R}, G, \mathbf{B})$. All these symbols are summarized in Table 4.1.

In our research, we have gathered two real world datasets to investigate the brand effect on item adoptions. The same datasets will also be used in our subsequent experiments (See Section 4.5). Detailed statistics of both datasets are provided in Table 4.2.

4.3.1 Real-world Datasets

To represent adoption data, we use *binary* ratings $\{r_{u,i}\}$ where $r_{u,i} = 1$ if user u adopts item i, $r_{u,i} = 0$ if u chooses not to adopt i and $r_{u,i} = undefined$ otherwise, e.g., u is not aware of the existence of i. It is noteworthy that non-existence of adoption record of user u for item i does NOT immediately imply that u chooses

Table 4.1: Symbols used in this paper

Symbol	**Description**
U ($\lvert U\rvert = N$)	Set of users
I ($\lvert I\rvert = M$)	Set of items
Q	Number of brands
K	Number of latent topics
$\mathbf{R} = (r_{u,i})_{U,I}$	Rating matrix
$G = (U, \mathbf{W})$	Directed, weighted network among users
$\mathbf{B}$	Matrix of brand-create-item relationships
$\mathscr{D} = (\mathbf{R}, G, \mathbf{B})$	Standard representation of a dataset
θ_u and β_u	Topic and brand factors of user u
θ_i and β_i	Topic and brand factors of item i
$w^{\mathbf{t}}_{u,v}$, $w^{\mathbf{b}}_{u,v}$ and $w_{u,v}$	Topic-based, brand-based and total influence of u on v
$r^{\mathbf{t}}_{u,i}$, $r^{\mathbf{b}}_{u,i}$ and $r_{u,i}$	Topic-based, brand-based and total rating of u for i
δ_u	Topic dependency weight of u
$\delta_U = (\delta_u)_{u \in U}$	*Topic dependency* vector of users

not to adopt i. In fact, there are two possibilities, namely: (i) u does NOT know about i OR (ii) u actually knows about i but chooses not to adopt. While we should assign 0 to $r_{u,i}$ in the latter case, an *undefined* value should be assigned to $r_{u,i}$ in the former case. However, it is not straightforward to distinguish the two cases. To mitigate this issue, we propose a proximity-based heuristics to reasonably impute 0's to ratings. The idea is that we only assign 0 to an item j with no adoption record from u when j is close enough to another item i already adopted by u (thus most likely "known" by u). This heuristics will be described in detail for each dataset.

Dataset from Foursquare (4SQDB). Foursquare is a popular location-based social network (LBSN) which allows users and venues to interact with one another. Users can follow other users. The follow network is represented as a directed graph $G = (U, \mathbf{W})$, where the weights in $\mathbf{W}$ are binary: $w_{u,v} = 1$ if v follows u, and 0 otherwise. Users can perform *check-ins* on venues as they visit the venues. Users

Table 4.2: Statistics of datasets

Dataset	# users (N)	# items (M)	# brands (Q)	# ratings	# edges
4SQDB	$4,940$	$14,821$	6237	$101,680$	$85,188$
ACMDB	$163,511$	$299,724$	$22,294$	$1,577,948$	$922,979$

can also write tips (a kind of short review) on venues. In our experiments, we consider a venue as an item. A brand in this dataset is a restaurant chain of food venue(s) (e.g., KFC), or just the food venue if it does not belong to any chain. When a user u checks in at a venue i, u is said to adopt i, i.e., the rating $r_{u,i} = 1$.

We collected raw check-in data on Foursquare from June 2011 to October 2015 via tweets of Singapore users, which were determined by the latitude and longitude of the tweets. In our experiments, we focus on check-ins only to *food venues* because of two reasons: (i) food venues contribute the largest number of check-ins compared to other kinds of venues, and (ii) focusing on only one type of venues lead to more interpretable results, especially for the subsequent evaluation of brand-conscious users. We also filtered out low-activity users with less than 3 check-ins and venues adopted by these users. After this filtering, we are left with 4940 users, each of them adopted at least 3 of $14,821$ items, i.e., food venues.

To impute 0-ratings into the rating matrix $\mathbf{R}$, we first divide all venues into 50 meters × 50 meters grid cells. For each pair (u,i) with $r_{u,i} = 1$, we randomly sample a venue j from other venues in the same cell such that u has not checked in and assign $r_{u,j} = 0$. All remaining $r_{u,i}$'s which are neither 1 nor 0 are assigned "undefined". In other words, we assume that a user checking into a venue i but not the nearby venue j has no interest in j. After this, we obtain the final dataset in standard representation $\mathscr{D} = (\mathbf{R}, G, \mathbf{B})$, which we denote as **4SQDB**.

Dataset from ACM Digital Library (ACMDB). From ACM Digital Library (ACMDL), we extract data of citations from 1998 to 2010. Each publication record consists of (i) its Id, title and abstract; (ii) its authors; and (iii) reference records; each of which contains Id, title and authors of a cited paper. Each citation is considered as an *adoption* where the cited paper is the adopted *item*. We then consider the *first author* of the citing paper as the *user* who adopts the item. The social network among the users is the weighted co-author network G where the weight on each edge (u,v) is the number of papers co-authored by u and v normalized by their total number of papers.

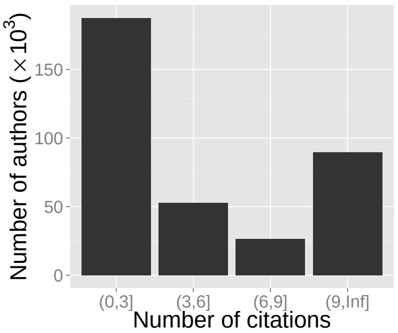


Figure 4.1: Distribution of citation count in original ACMDL data

As the total number of authors, around 350K, is huge, it is not feasible to consider all of them as brands. Instead, we empirically select those authors with citation count significantly higher than that of a normal author. We plot the distribution of citation count (Figure 4.1) in original data and find that (i) the median number of citations is 3, and (ii) about 25% of authors have at least 9 citations, which is 3 times more than the median. Thus, we decide that only authors with at least 9 citations should be considered as brands. We thus retain only those authors as *brands* and extract (i) their items to form the item set I and the brand-create-item matrix $\mathbf{B}$, and (ii) users who adopt at least one of these items to form the set of users U. The co-author network among the users is thus a sub network of the original G, we however still denote it as G for simplicity.

Again, each adoption is a rating with value 1. We then impute 0s to obtain the rating matrix $\mathbf{R}$ by the proximity-based heuristics as what we did on Foursquare data. However, the similarity between two papers is now defined by the Jaccard similarity between two keyword sets extracted from their abstracts. After this, we obtain the final dataset in standard representation $\mathscr{D} = (\mathbf{R}, G, \mathbf{B})$, which we denote as **ACMDB**.

4.3.2 Empirical Analysis

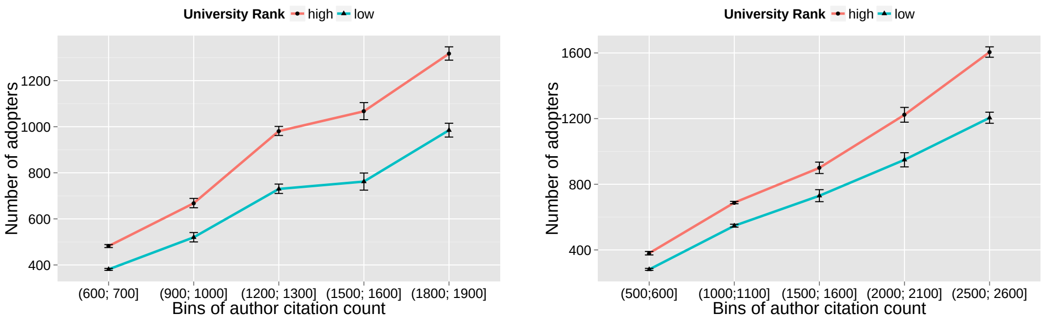
We now perform two kinds of empirical analysis on the datasets. The first investigates into the existence of brand effect on user-item adoption. We would like to validate the assumption of brand affecting the choice of items adopted by users, especially for some well known brands. The second examines the correlation between social tie strength and user-user brand similarity. This correlation study will help to validate the assumption that social tie also plays a role in users' brand choices.

Analysis of Brand Effect

As we do not have a direct measure for brand value of an author, we use the brand value of the university where the author works as a proxy. We would like to check if authors from high-rank universities *attract more attention* than those from low-rank universities, controlling for the author's research topics, citation counts and country of the university affiliation. We measure the amount of attention received by an author by the number of adopters of her papers. We thus create a dataset of citation counts and adopter counts of authors from US universities by combining various data sources as follows.

- We extracted the citation count and adopter count of authors from ACMDL. An author's citation count and adopter count refer to the number of papers and number of authors citing the papers of the author respectively.
- Author affiliation and topic data were obtained from the author profiles in Google Scholar. As the number of authors we could crawl is limited to about 550 per machine, we only crawled the profiles of authors under two research topics, namely, *data mining* and *distributed systems*. These profiles are found by querying Google Scholar with appropriate query terms and extracting the required fields from the returned author profile results. To query authors under data mining, we used the terms "data mining", "text mining", and "social network mining". To query authors under distributed systems, we used the terms "distributed systems", "concurrent", and "parallel". We also performed a manual check on retrieved authors to remove some exceptions outside the two topics.
- University ranking in the Computer Science discipline was crawled from the

Figure 4.2: High-rank-university vs. low-rank-university cited authors: a comparison on adopter count.



(a) Authors under “distributed systems”

(b) Authors under “data mining”

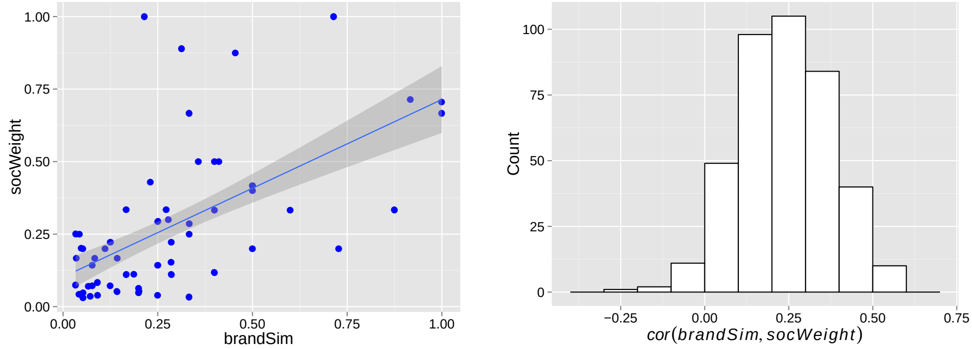
website *www.topuniversities.com*. We obtained from the website ranks of 90 US universities. Empirically, we consider universities with ranks 1-45 (46-90) as high (low) rank universities.

For each topic, we then assign authors into bins by their number of citations such that each bin has 40-50 authors. Figure 4.2 depicts the adopter counts of authors of both high rank and low rank universities in different citation count bins. To keep the figure simple, we only show five citation count bins in each chart. By analyzing adopter counts of authors in each bin, we found that authors from high rank universities indeed have more adopters than those from low rank universities (see Figures 4.2a and 4.2b). This difference can be found for the authors under both “data mining” and “distributed systems” topics. The difference is smaller (e.g., around 100 for the (600,700] bin for the “distributed systems” topic) for authors with smaller citation counts but larger (e.g., around 300 for the (1800,1900] bin for the “distributed systems” topic) for authors with larger citation counts. This confirms the existence of brand effect.

Analysis of Social Correlation

Users who are alike choose to be connected with one another, and socially connected users influence one another to be even more similar. These two processes in social networks, known as *selection* and *social influence*, are expected to affect

Figure 4.3: Correlation between social tie weight and brand-based similarity, observed on citation data from ACMDL



(a) Scatter plot with regression line for a sample of 1000 users with 118 pairs of co-authors having brand similarity (each point is one pair). The sample's correlation value is 0.5

(b) Empirical distribution of the correlation between social tie weight and brand similarity. Correlation values are computed on 400 samples, each consists of 1000 users.

connected users' topic preferences and brand preferences [40, 94**?**]. Research has shown that recommendation methods modeling social network effect on users' topic preferences can achieve better recommendation accuracy [24, 60, 85, 86]. Inspired by these results, we analyzed ACMDB to confirm a positive correlation between *social tie weight* and user similarity in brand preference, which we call *user-user brand similarity*.

Firstly, we computed the social tie weight $socWeight(u,v)$ as the number of papers co-authored by u and v normalized by their total number of papers:

$$socWeight(u,v) = \frac{|P(u) \cap P(v)|}{|P(u) \cup P(v)|}$$

where $P(x)$ are the papers of which x is one of the authors. We then extracted user pairs who adopted at least one common "brand", i.e., cited at least one common author, and calculated brand similarity for each pair based on authors adopted by both users:

$$brandSim(u,v) = \frac{|A(u) \cap A(v)|}{|A(u) \cup A(v)|}$$

where $A(x)$ are the authors cited by x.

Finally, we computed the Pearson correlation between the user-user brand similarity and the social tie weight. Computing this correlation on the whole co-author network is costly as the network is large with nearly $200,000$ users and millions of edges. Thus, we resorted to computing the correlation on samples of randomly chosen users, each of size 1000 users. We iterated this process for 400 samples and plotted the histogram of the correlation values in Figure 4.3b. The figure shows that most correlations are positive and their mean is 0.23. This suggests a positive correlation between user-user brand similarity and social tie weight.

4.3.3 Proposed Models

In our proposed models SocBIT and SocBIT$^+$, we introduce a few important definitions and assumptions. Let B denote the set of all brands in data. We first formally define user brand preference as a vector of numeric factors, one for each brand. The factor of each brand indicates how much a user prefers the brand. In the case of SocBIT$^+$, each user-brand factor is non-negative and the larger it is, the more the user prefers the brand.

Definition 7 (Brand factors of user)**.** *Given a user u, the brand factor $\beta_{b,u}$ of u for a brand $b \in \mathsf{B}$ measures u's preference toward brand b. In the context of SocBIT$^+$, we have $\beta_{b,u} \geq 0, \forall b$.*

Next, we define brand factors of item. This arises from the remark that different items of the same brand represent the brand differently, which affects the ratings they receive from users. For example, (i) only signature dishes of a restaurant are the most representative ones and thus preferred by its customers over normal dishes; (ii) among movies of Jackie Chan, the more representative ones receive higher ratings from audience. In other words, for each brand b and each item i that the brand creates, there should be a latent factor measuring the extent to which i represents b. We integrate this into our models by defining this factor as the *brand factor* of item i with respect to brand b. When an item is not created by a brand, the corresponding item-brand factor is 0. In short, we associate each item with a vector of item-brand factors as follows.

Definition 8 (Brand factors of item)**.** *Given an item i and the set B_i of brands creating i, the brand factor $\beta_{b,i}$ of item i for brand $b \in \mathsf{B}_i$ measures the extent to which i represents b. For $b \notin \mathsf{B}_i$, i.e. brand b does not create item i, $\beta_{b,i}$ is simply 0. In the context of SocBIT$^+$, we have $\beta_{b,i} \geq 0, \forall b \in \mathsf{B}$.*

For a user u to adopt an item i based on brand, u and i should have high user-item brand-based similarity, which can be measured by the sum $\sum_{b \in \mathsf{B}_i} \beta_{u,b}\beta_{i,b}$. When u adopts an i based on topic, the user-item topic similarity will be used as in matrix factorization models.

Assumption 1 (Topic-based and Brand-based Ratings)**.** *The rating a user gives to an item can be approximated as a weighted average of topic-based and to brand-based similarities. The weight is user dependent.*

Our analysis of social correlation in Section 4.3.2 suggests that user-user brand similarity is higher for users with stronger social ties. By combining the correlation of topics and brands among socially connected users, we propose the second assumption.

Assumption 2 (Social Correlation)**.** *Social tie strength between any two users **correlates** with their topic-based and brand-based similarities.*

Given these assumptions, we now formulate SocBIT model based on matrix factorization framework. We first describe how SocBIT jointly models the generative processes of ratings and social weights. We then propose SocBIT's conditional probabilities based on the generative processes. Finally, we derive SocBIT's posterior from the conditional probabilities and Gaussian priors.

Generative Process

The plate diagram of SocBIT is given in Figure 4.4. The left plate represents users and how social connections among them are generated. The right plate represents items and how ratings are generated. Similar to traditional MF, each user u and item i is assigned a user topic vector θ_u and item topic vector θ_i respectively. Both vectors are K dimensional w.r.t. K topics. By Definition 7, SocBIT models brand preference of user u by a vector $\beta_u = (\beta_{b,u})_{b \in \mathsf{B}}$. By Definition 8, item i is associated with a vector $\beta_i = (\beta_{b,i})_{b \in \mathsf{B}}$, where each element $\beta_{b,i}$ measures how much i represents brand b. Both β_u and β_i vectors are Q-dimensional corresponding to the number of brands. Finally, when i does not belong to brand b, i.e., the entry $B_{i,b}$ in brand-create-item matrix $\mathbf{B}$ is 0, the factor $\beta_{b,i}$ is also 0.

SocBIT models the generation process of each rating $r_{u,i}$ as follows. First, the topic vectors of u and i are used to generate a **topic-based** rating $r^{\mathbf{t}}_{u,i}$. Meanwhile, the brand vectors of u and i are used to generate a **brand-based** rating $r^{\mathbf{b}}_{u,i}$. By Assumption 1, the final rating $r_{u,i}$ is then approximated as a weighted average of $r^{\mathbf{t}}_{u,i}$ and $r^{\mathbf{b}}_{u,i}$. The weights depend on how much user u depends on topics to assign ratings, which is measured by the *topic dependency weight* $\delta_u \in [0,1]$. In short, by defining the ratings $r^{\mathbf{t}}_{u,i}, r^{\mathbf{b}}_{u,i}$ as

$$r^{\mathbf{t}}_{u,i} = \theta_u^T \theta_i, \quad r^{\mathbf{b}}_{u,i} = \beta_u^T \beta_i$$

we can approximate $r_{u,i}$ by

$$r_{u,i} \approx \widehat{r}_{u,i} \overset{\text{def}}{=} g\left(\delta_u r^{\mathbf{t}}_{u,i} + (1-\delta_u) r^{\mathbf{b}}_{u,i}\right) = g(\overbrace{\delta_u \theta_u^T \theta_i + (1-\delta_u)\beta_u^T \beta_i}^{\gamma_{u,i}}) = g(\gamma_{u,i}) \quad (4.3)$$

where the logistic function $g(x) = 1/(1+e^{-x})$ is used to bound the approximated

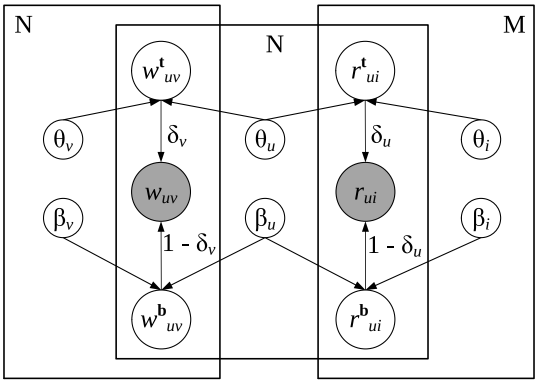


Figure 4.4: Graphical model for SocBIT

rating in $[0,1]$.

Other than observed ratings, SocBIT also models observed social network connections $\mathbf{W}$. The left plate in Figure 4.4 denotes the users u has influence over. SocBIT assumes the social weight u exerts on v is accounted by two similarities between them. The first is topic-based similarity $w^{\mathbf{t}}_{u,v}$ and the second is brand-based similarity $w^{\mathbf{b}}_{u,v}$. The final social weight is thus a weighted average of $w^{\mathbf{t}}_{u,v}$ and $w^{\mathbf{b}}_{u,v}$ as follows.

$$w_{u,v} \approx \widehat{w}_{u,v} \overset{\text{def}}{=} g(\overbrace{\delta_v \theta_u^T \theta_v + (1-\delta_v)\beta_u^T \beta_v}^{\omega_{u,v}}) = g(\omega_{u,v}) \tag{4.4}$$

Conditional Probabilities

In this section, we propose the conditional probabilities for rating and social network matrices. For that, we first need to rewrite Equations (4.3) and (4.4) in matrix factorization form. As the form show how the two matrices are estimated given parameters of SocBIT, it is then straightforward to obtain the conditional probabilities. We need some more matrix notations.

- $\Theta_U = (\theta_u)_U \in \mathbb{R}^{K\times N}$ and $\Theta_I \in \mathbb{R}^{K\times M}$: matrices of topic factors of users and items,
- $\mathscr{B}_U = (\beta_u)_U \in \mathbb{R}^{Q\times N}$ and $\mathscr{B}_I = (\beta_i)_I \in \mathbb{R}^{Q\times M}$: matrices of brand factors of users and items,
- $\Delta_U = diag(\delta_u)_U \in \mathbb{R}^{N\times N}$: diagonal matrix of which diagonal entries are topic dependency weights of users,
- π= $(\Theta_U, \Theta_I, \mathscr{B}_U, \mathscr{B}_I, \delta_U)$: all the parameters of SocBIT.

Conditional Probability for Ratings: We now can rewrite Equation (4.3) as

$$\mathbf{R} \approx \widehat{\mathbf{R}} \overset{\text{def}}{=} g\left(\Delta_U \Theta_U^T \Theta_I + (\mathbf{Id} - \Delta_U)\, \mathscr{B}_U^T \mathscr{B}_I\right) \tag{4.5}$$

where $g(\mathbf{A})$ of a matrix $\mathbf{A}$ is simply the matrix obtained by applying the logistic function on $\mathbf{A}$ element-wise and $\mathbf{Id}$ denotes the identity matrix. This form inspires the conditional probability

$$p(\mathbf{R}|\pi;\sigma_R) = \prod_{(u,i)} \mathcal{N}(r_{u,i}|\widehat{r}_{u,i}(\underbrace{\delta_u,\theta_u,\theta_i,\beta_u,\beta_i}_{\pi_{u,i}});\sigma_R^2)^{\mathbb{1}_{u,i}^R} = \prod_{(u,i)} \mathcal{N}\left(r_{u,i}|\widehat{r}_{u,i}(\pi_{u,i});\ \sigma_R^2\right)^{\mathbb{1}_{u,i}^R} \tag{4.6}$$

where $\mathbb{1}_{u,i}^R$ is the indicator on whether u actually rates i.

One may argue that the decomposition in Equation (4.5) may be problematic as matrices $\mathscr{B}_U$ and $\mathscr{B}_I$ have high dimension Q. However, in reality, these matrices have lots of 0 entries as each user is only interested in a few brands and each item only belongs to a few brands. Thus, the product $\mathscr{B}_U^T\mathscr{B}_I$ is still equivalent to a low rank factorization.

Conditional probability for Social Weights: Similarly, Equation (4.4) can be rewritten as follows.

$$\mathbf{W} \approx \widehat{\mathbf{W}} \overset{\text{def}}{=} g\left(\Delta_U\Theta_U^T\Theta_U + (\mathbf{Id}-\Delta_U)\mathscr{B}_U^T\mathscr{B}_U\right) \tag{4.7}$$

The conditional probability for social matrix $\mathbf{W}$ is then:

$$\begin{aligned} p(\mathbf{W}|\pi,\sigma_W) &= \prod_{u,v} \mathcal{N}(w_{u,v}|\widehat{w}_{u,v}(\underbrace{\delta_v,\theta_u,\theta_v,\beta_u,\beta_v}_{\pi_{u,v}});\sigma_W^2)^{\mathbb{1}_{u,v}^W} \\ &= \prod_{u,v} \mathcal{N}(w_{u,v}|\widehat{w}_{u,v}(\pi_{u,v});\sigma_W^2)^{\mathbb{1}_{u,v}^W} \end{aligned} \tag{4.8}$$

where $\mathbb{1}_{u,v}^W$ is the indicator on whether there is a social tie from u to v.

Posterior of SocBIT

Similar to SoRec [85], we use spherical Gaussian distribution as priors. In the following, the standard deviations of the Gaussian distributions for *user-topic, user-brand, item-topic, item-brand* factors and *topic dependency weights* are denoted as $\sigma_{\mathbf{ut}}, \sigma_{\mathbf{ub}}, \sigma_{\mathbf{it}}, \sigma_{\mathbf{ib}}$ and $\sigma_{\mathbf{d}}$ respectively.

- (Priors for user factors)

$$p(\Theta_U|\sigma_{\mathbf{ut}}) = \prod_{u \in U} \mathscr{N}(\theta_u|\mathbf{0}_K; \sigma^2_{\mathbf{ut}}\mathbf{Id}) \text{ and } p(\mathscr{B}_U|\sigma_{\mathbf{ub}}) = \prod_{u \in U} \mathscr{N}(\beta_u|\mathbf{0}_Q; \sigma^2_{\mathbf{ub}}\mathbf{Id}) \tag{4.9}$$

- (Priors for item factors)

$$p(\Theta_I|\sigma_{\mathbf{it}}) = \prod_{i \in I} \mathscr{N}(\theta_i|\mathbf{0}_K; \sigma^2_{\mathbf{it}}\mathbf{Id}) \text{ and } p(\mathscr{B}_I|\sigma_{\mathbf{ib}}) = \prod_{i \in I} \mathscr{N}(\beta_i|\mathbf{0}_Q; \sigma^2_{\mathbf{ib}}\mathbf{Id}) \tag{4.10}$$

- (Prior for topic dependency weights) We assume each δ_u has mean 0.5 as most people are neither brand-conscious nor non-brand-conscious.

$$p(\delta_U|\sigma_{\mathbf{d}}) = \prod_{u \in U} \mathscr{N}(\delta_u|0.5, \sigma^2_{\mathbf{d}}) \tag{4.11}$$

Given these conditional probabilities and priors, the joint probability is then obtained as

$$\begin{aligned} P(\mathbf{W}, \mathbf{R}, \pi|\Sigma) = & [p(\mathbf{W}|\pi; \sigma_W) p(\mathbf{R}|\pi; \sigma_R)] \times \\ & [p(\Theta_U|\sigma_{\mathbf{ut}}) p(\mathscr{B}_U|\sigma_{\mathbf{ub}}) p(\Theta_I|\sigma_{\mathbf{it}}) p(\mathscr{B}_I|\sigma_{\mathbf{ib}}) p(\delta_U|\sigma_{\mathbf{d}})] \end{aligned} \tag{4.12}$$

where $\Sigma = (\sigma_W, \sigma_R, \sigma_{\mathbf{ut}}, \sigma_{\mathbf{ub}}, \sigma_{\mathbf{it}}, \sigma_{\mathbf{it}}, \sigma_{\mathbf{d}})$ represents all hyper-parameters.

By Bayes theorem, SocBIT's posterior is proportional to its joint probability. Thus, we obtain the negative log posterior as follows.

$$\begin{aligned} -\ln P(\pi|\mathbf{W}, \mathbf{R}; \Sigma) \propto & \frac{1}{\sigma_R^2} \sum_{(u,i)} \mathbb{1}^R_{u,i} [\widehat{r}_{u,i}(\pi_{u,i}) - r_{u,i}]^2 + \frac{1}{\sigma_W^2} \sum_{(u,v)} \mathbb{1}^W_{u,v} [\widehat{w}_{u,v}(\pi_{u,v}) - w_{u,v}]^2 + \\ & \frac{1}{\sigma^2_{\mathbf{ut}}} \|\Theta_u\|_F^2 + \frac{1}{\sigma^2_{\mathbf{ub}}} \|\mathscr{B}_u\|_F^2 + \frac{1}{\sigma^2_{\mathbf{ut}}} \|\Theta_i\|_F^2 + \frac{1}{\sigma^2_{\mathbf{ib}}} \|\mathscr{B}_i\|_F^2 + \\ & \frac{1}{\sigma^2_{\mathbf{d}}} \|\delta_U - 0.5\|^2 \end{aligned} \tag{4.13}$$

where $\|.\|_F$ denotes the usual Frobenius norm of matrix.

4.3.4 SocBIT Inference

Maximizing the log posterior over model parameters is equivalent to minimizing the following squared-error objective function with quadratic regularization terms.

$$\begin{aligned}\mathscr{L}(\pi) = \frac{1}{2}\sum_{(u,i)} \mathbb{1}^R_{u,i}\left[\widehat{r}_{u,i}(\pi_{u,i}) - r_{u,i}\right]^2 + \frac{\lambda_W}{2}\sum_{(u,v)}\left[\widehat{w}_{u,v}(\pi_{u,v}) - w_{u,v}\right]^2 + \\ \frac{\lambda^{\mathbf{t}}_U}{2}\|\Theta_U\|^2_F + \frac{\lambda^{\mathbf{b}}_U}{2}\|\mathscr{B}_U\|^2_F + \frac{\lambda^{\mathbf{t}}_I}{2}\|\Theta_I\|^2_F + \frac{\lambda^{\mathbf{b}}_I}{2}\|\mathscr{B}_I\|^2_F + \frac{\lambda_{\mathbf{d}}}{2}\|\delta_U - 0.5\|^2\end{aligned} \tag{4.14}$$

where regularization coefficients are

$$\{\lambda_W, \lambda_{\mathbf{d}}, \lambda^{\mathbf{t}}_U, \lambda^{\mathbf{b}}_U, \lambda^{\mathbf{t}}_I, \lambda^{\mathbf{b}}_I\} = \sigma^2_R\{1/\sigma^2_W, 1/\sigma^2_{\mathbf{d}}, 1/\sigma^2_{\mathbf{ut}}, 1/\sigma^2_{\mathbf{ub}}, 1/\sigma^2_{\mathbf{it}}, 1/\sigma^2_{\mathbf{ib}}\}.$$

To reduce model complexity and avoid overfitting, we set $\lambda^{\mathbf{t}}_U = \lambda^{\mathbf{t}}_I = \lambda_{\mathbf{t}}$ and $\lambda^{\mathbf{b}}_U = \lambda^{\mathbf{b}}_I = \lambda_{\mathbf{b}}$.

A local minimum of this objective function can be found by performing projected gradient descent on the model parameters. The projection is needed to ensure that $\beta_{b,i} = 0$ when $B_{b,i} = 0$. We now show formulae of the gradients.

Gradients for item factors. When we derive the gradients of objective function for a given item i, the second term will vanish as it does not involve items. Thus, each of the gradients w.r.t. θ_i and β_i depends on only two components: (i) rating estimations, and (ii) regularizers. We thus have

$$\nabla_{\theta_i}\mathscr{L} = \lambda^{\mathbf{t}}_I\theta_i + \sum_{u\in U}\delta_u\mathbb{1}^R_{u,i}(\widehat{r}_{u,i} - r_{u,i})g'(\gamma_{u,i})\theta_u \tag{4.15}$$

and

$$\nabla_{\beta_i}\mathscr{L} = \lambda^{\mathbf{b}}_I\beta_i + \sum_{u\in U}(1-\delta_u)\mathbb{1}^R_{u,i}(\widehat{r}_{u,i} - r_{u,i})g'(\gamma_{u,i})\beta_u \tag{4.16}$$

Gradients for user factors. For a given user u, each of the gradients w.r.t θ_u

and β_u depends on three components: (i) rating estimations, (ii) weight estimations, and (iii) regularizers. Thus, we have

$$\begin{aligned}\nabla_{\theta_u}\mathscr{L} =& \lambda_U^{\mathbf{t}}\theta_u + \\ & \delta_u\left[\sum_{i\in I}\mathbb{1}_{u,i}^R(\widehat{r}_{u,i}-r_{u,i})g'(\gamma_{u,i})\theta_i + \lambda_W\sum_{v\in U}\mathbb{1}_{u,v}^W(\widehat{w}_{u,v}-w_{u,v})g'(\omega_{u,v})\theta_v\right]\end{aligned} \tag{4.17}$$

and

$$\begin{aligned}\nabla_{\beta_u}\mathscr{L} =& \lambda_U^{\mathbf{b}}\beta_u + \\ & (1-\delta_u)\left[\sum_{i\in I}\mathbb{1}_{u,i}^R(\widehat{r}_{u,i}-r_{u,i})g'(\gamma_{u,i})\beta_i + \lambda_W\sum_{v\in U}\mathbb{1}_{u,v}^W(\widehat{w}_{u,v}-w_{u,v})g'(\omega_{u,v})\beta_v\right]\end{aligned} \tag{4.18}$$

Derivatives for topic dependency weights. For a given user u, the derivative of the objective function for topic dependency weight δ_u is:

$$\begin{aligned}\frac{\partial\mathscr{L}}{\partial\delta_u} = & \lambda_{\mathbf{d}}(\delta_u - 0.5) + \sum_{i\in I}\mathbb{1}_{u,i}^R(\widehat{r}_{u,i}-r_{u,i})(\theta_u^T\theta_i - \beta_u^T\beta_i)g'(\gamma_{u,i}) - \\ & \lambda_W\sum_{v\in U}\mathbb{1}_{u,v}^W(\widehat{w}_{u,v}-w_{u,v})(\theta_u^T\theta_v - \beta_u^T\beta_v)g'(\omega_{u,v})\end{aligned} \tag{4.19}$$

4.3.5 Nonnegative Version - SocBIT$^+$

One issue of the GD inference is that its *additive* update rules cannot guarantee the non-negativity of user and item factors. Using the approach in [71], we replace additive update rules by *multiplicative* update rules, which grants us the desired non-negativity. The core idea of the approach is choosing suitable step size to cancel out negative parts in update formulae as follows.

We start with the gradient for updating item topic factors in Equation (4.15).

The corresponding update formula, in element-wise form, is then

$$\theta_{i,k} \leftarrow \theta_{i,k} - \eta_{i,k}\left[\lambda_{\mathbf{t}}\theta_{i,k} + \sum_{u\in U}\mathbb{1}^{R}_{u,i}\delta_u g'(\gamma_{u,i})\widehat{r}_{u,i}\theta_{u,k}\right] + \eta_{i,k}\left(\sum_{u\in U}\mathbb{1}^{R}_{u,i}\delta_u g'(\gamma_{u,i})r_{u,i}\theta_{u,k}\right) \tag{4.20}$$

To cancel out the negative part, we need

$$\theta_{i,k} - \eta_{i,k}\left[\lambda_{\mathbf{t}}\theta_{i,k} + \sum_{u\in U}\mathbb{1}^{R}_{u,i}\delta_u g'(\gamma_{u,i})\widehat{r}_{u,i}\theta_{u,k}\right] = 0$$

The proper step size is then

$$\eta_{i,k} = \frac{\theta_{i,k}}{\lambda_{\mathbf{t}}\theta_{i,k} + \sum_{u\in U}\mathbb{1}^{R}_{u,i}\delta_u g'(\gamma_{u,i})\widehat{r}_{u,i}\theta_{u,k}}$$

With this value of $\eta_{i,k}$, the negative part is cancelled and only the last term in Equation (4.20) remains. Thus, we obtain the following multiplicative update rule.

Updating topic factor k of item i:

$$\theta_{i,k} \leftarrow \theta_{i,k} \times \frac{\sum_{u\in U}\mathbb{1}^{R}_{u,i}\delta_u g'(\gamma_{u,i})r_{u,i}\theta_{u,k}}{\lambda_{\mathbf{t}}\theta_{i,k} + \sum_{u\in U}\mathbb{1}^{R}_{u,i}\delta_u g'(\gamma_{u,i})\widehat{r}_{u,i}\theta_{u,k}} \tag{4.21}$$

We can proceed similarly to obtain the following update rules for the remaining user and item factors.

Updating brand factor b of item i:

$$\beta_{i,b} \leftarrow \beta_{i,b} \times \frac{\sum_{u\in U}\mathbb{1}^{R}_{u,i}(1-\delta_u)g'(\gamma_{u,i})r_{u,i}\beta_{u,b}}{\lambda_{\mathbf{b}}\beta_{i,b} + \sum_{u\in U}\mathbb{1}^{R}_{u,i}(1-\delta_u)g'(\gamma_{u,i})\widehat{r}_{u,i}\beta_{u,b}} \tag{4.22}$$

Updating topic factor k of user u:

$$\theta_{u,k} \leftarrow \theta_{u,k} \times \frac{\delta_u\left[\sum_{i\in I}\mathbb{1}^{R}_{u,i}g'(\gamma_{u,i})r_{u,i}\theta_{i,k} + \sum_{v\in U}\mathbb{1}^{W}_{u,v}g'(\omega_{u,v})w_{u,v}\theta_{v,k}\right]}{\lambda_{\mathbf{t}}\theta_{u,k} + \delta_u\left[\sum_{i\in I}\mathbb{1}^{R}_{u,i}g'(\gamma_{u,i})\widehat{r}_{u,i}\theta_{i,k} + \sum_{v\in U}\mathbb{1}^{W}_{u,v}g'(\omega_{u,v})\widehat{w}_{u,v}\theta_{v,k}\right]} \tag{4.23}$$

Updating brand factor b of user u:

$$\beta_{u,b} \leftarrow \beta_{u,b} \times \frac{(1-\delta_u)\left[\sum_{i\in I} \mathbb{1}^R_{u,i} g'(\gamma_{u,i}) r_{u,i}\beta_{i,b} + \sum_{v\in U} \mathbb{1}^W_{u,v} g'(\omega_{u,v}) w_{u,v}\beta_{v,b}\right]}{\lambda_{\mathbf{b}}\beta_{u,b} + (1-\delta_u)\left[\sum_{i\in I} \mathbb{1}^R_{u,i} g'(\gamma_{u,i}) \widehat{r}_{u,i}\beta_{i,b} + \sum_{v\in U} \mathbb{1}^W_{u,v} g'(\omega_{u,v}) \widehat{w}_{u,v}\beta_{v,b}\right]} \tag{4.24}$$

To update the topic dependency weights δ_u's, we still use Equation (4.19). As long as δ_u's remain in $[0,1]$, all these update rules will guarantee the non-negativity of user and item factors. We thus ensure this by applying cut-off to bring any δ_u outside $[0,1]$ back to the range.

4.4 Experiments on Synthetic Data

The experiments in this section examine performance of our models given that the underlying assumptions (see Section 4.3.3) are satisfied. Specifically we aim to address the following questions.

1. Do our models outperform state-of-the-art recommendation models in adoption prediction task?
2. Can our models recover ground truth parameters correctly?
3. If brand-conscious users exist in data, can our models detect them?

In the following sections, we first address Question 1 in Section 4.4.3. We then address question 2 in Section 4.4.4. Finally question 3 is addressed in Section 4.4.5.

Table 4.3: Parameters for synthetic experiments

Symbol	Description	Range (**default value**)
φ_b	Fraction of brand-conscious users	$[0,0.5]$ (**0.1**)
N	Number of users	**10000**
M	Number of items	**1000**
Q	Number of brands	**50**
K	Number of topics	**5**

4.4.1 Data Generation

In this section, we elaborate on the process of generating a synthetic dataset $\mathscr{D} = (\mathbf{R}, G, \mathbf{B})$. To examine the performance of our models w.r.t. different fractions φ_b of brand-conscious users in data, we generate several synthetic datasets for different φ_b's. However, the numbers of topics, users, items and brands are fixed across all these datasets for a fair comparison. Thus, each dataset consists of $N = 10000$ users, $K = 5$ topics, $Q = 50$ brands and $M = 1000$ items. For easy reading, we also provide a summary of parameters needed for synthetic data experiments in Table 4.3.

To facilitate fair comparison with other models, we do not bind the generation process rigidly to SocBIT's formulation. Instead, we just let it simulate the underlying assumptions of our proposed approach, namely *homophily* and *brand-affected* rating decision (Section 4.3.3). The process sequentially generates: (i) brands, (ii) items, (iii) brand-create-item matrix $\mathbf{B}$, (iv) users, and (v) rating matrix $\mathbf{R}$ and network G as follows.

Brand Generation:

To be precise, for each topic k, we generate the set of brands which are *relevant* to k, i.e., these brands will create items under topic k, and denote the set as B_k. For any two topics, say k and j, we allow them to have common brands but keep the size of the overlap no more than 10% of the minimum number of brands of the two topics. This can be done by assigning consecutively to each topic $10q = 10\lfloor Q/(9K+1) \rfloor$ brands s.t. any two consecutive topics has q overlapping brands.

For each topic k, we then assign randomly one of its brands as the most popular brand, denoted as b_k^* and forms k's brand distribution having a single mode at b_k^* with probability 0.8 and uniform probability $0.2/(10q-1)$ for each of its remaining brands. This process generates a $K \times Q$ topic-brand matrix $\mathscr{Z} = (\zeta_k)_{k=1,K}$ where the k-th row is the brand distribution representing popularities of brands under topic k. This matrix will be needed later in item and user generation.

Item Generation:

We choose M to be a multiple of Q so that we can partition M items into Q disjoint subsets, each of size $m = M/Q$. We then assign the b-th subset of items to brand b s.t. each item only belongs to a *single* brand. This will give us the brand-create-item matrix.

Moreover, for each brand, the first 10% of its items are considered as the ones with highest brand factor and they are assigned ground-truth brand factors of value 5 while the remaining are normal items with ground-truth brand factors of value 3.

If we denote

$$\alpha = (\underbrace{5,\ldots,5}_{\lfloor m/10 \rfloor},\underbrace{3,\ldots\ldots3}_{m-\lfloor m/10 \rfloor})$$

then the matrix $\mathscr{B}_I$ representing *brand factors* of items will be the following block diagonal matrix of dimensions $Q \times M$.

$$\mathscr{B}_I = \begin{bmatrix} \alpha & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \alpha \end{bmatrix} \tag{4.25}$$

Finally, we obtain the matrix of *topic factors* of items as $\Theta_I = \mathscr{X} \times \mathscr{B}_I$. Thus, we finish generating a reasonable set of items which possess both brand and topic factors. We now continue with user generation.

User Generation:

We generate a set of users embedded with a small proportion $\varphi_b (\leq 0.5)$ of brand-conscious users s.t. each user has his own topic and brand factors. The detailed process is described below.

1. Create a set of N users such that a fraction φ_b of them are brand-conscious. We use S_b to denote the set of brand-conscious users.

2. Assign to each user a topic dependency weight. The weights are sampled from $uniform([0,0.1])$ and from $uniform([0.5,1])$ for users in S_b and for

users outside S_b respectively. The weights form the diagonal matrix Δ_u needed later for rating and network generation.

3. Generating *topic factors* θ_u of u by (i) randomly assigning a topic f_u to be u's favorite topic, leaving $K-1$ remaining topics as non-favorite; (ii) assigning the highest preference score 0.7 to f_u and uniform preference score $0.3/(K-1)$ to all remaining topics.

4. Generate *brand factors* β_u of users as $\mathscr{B}_U = \mathscr{Z}^T \times \Theta_U$.

Rating and Network Generation:

Now that all factors of users and items have been assigned, we first apply the decomposition in Equation (4.5) on the factors to obtain a full rating matrix $\mathbf{R}_{full}$ of which all entries are defined. We then generate different rating matrices $\mathbf{R}$'s, each with a given percentage $\rho_{\mathbf{R}}$ of defined entries, by randomly retaining only a percentage ρ_R of entries in $\mathbf{R}_{full}$ and setting others to "undefined".

Finally, we generate a network among the users based on homophily principle, i.e. similar users are more likely to connect with each other. Specifically, the probability $p(\mathbb{1}(u,v)=1)$ of an edge being formed from user u to user v is simply proportional to the average of their topic and brand similarities $p(\mathbb{1}(u,v)=1) \propto g(\frac{\theta_u^T \theta_v + \beta_u^T \beta_v}{2})$. By combining these probabilities with a power-law degree distribution, we generate a binary adjacency matrix $\mathbf{W}$ of a synthetic graph G.

4.4.2 Metrics and Baseline

We use Root Mean Squared Error (RMSE) for evaluating the accuracy of models. Given a model, we define its RMSEs on a training set `train` and a test set `test` as follows.

Definition 9 (RMSEs). *Denote $\widehat{r}_{u,i}$ as the rating of user u on item i estimated by the model. We have*

$$RMSE_{train} \overset{def}{=} \sqrt{\frac{1}{N_{train}} \sum_{\mathtt{train}} (\widehat{r}_{u,i} - r_{u,i})^2} \tag{4.26}$$

and

$$RMSE_{test} \overset{def}{=} \sqrt{\frac{1}{N_{test}} \sum_{\mathtt{test}} (\widehat{r}_{u,i} - r_{u,i})^2} \tag{4.27}$$

where N_{train} and N_{test} are the numbers of ratings in training and test set respectively.

For the task of recovering ground truth parameters, we evaluate accuracy of the models based on the following error metrics.

Definition 10 (Topic Factor Recovery Errors)**.** *The errors of a given model in recovering user and item* topic factors*, denoted as $Err_{\mathbf{ut}}$ and $Err_{\mathbf{it}}$ respectively, are defined as average squared errors over users and items respectively*

$$Err_{\mathbf{ut}} \overset{def}{=} \sqrt{\frac{1}{N} \sum_{u \in U} (\widehat{\theta}_u - \theta_u)^2} \quad and \quad Err_{\mathbf{it}} \overset{def}{=} \sqrt{\frac{1}{M} \sum_{i \in I} (\widehat{\theta}_i - \theta_i)^2} \tag{4.28}$$

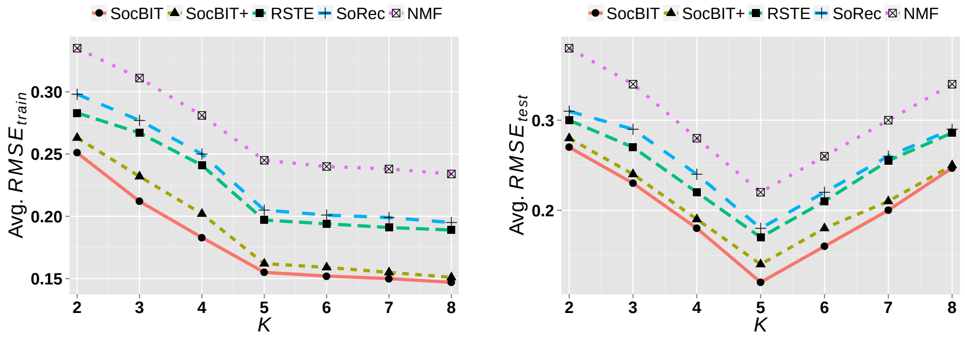
As SocBIT and SocBIT$^+$ also learn *brand factors* of users and items, we investigate their errors in recovering such brand factors, denoted as $Err_{\mathbf{ub}}$ and $Err_{\mathbf{ib}}$ respectively. The errors are defined as

Definition 11 (Brand Factor Recovery Errors)**.**

$$Err_{\mathbf{ub}} \overset{def}{=} \sqrt{\frac{1}{N} \sum_{u \in U} (\widehat{\beta}_u - \beta_u)^2} \quad and \quad Err_{\mathbf{ib}} \overset{def}{=} \sqrt{\frac{1}{M} \sum_{i \in I} (\widehat{\beta}_i - \beta_i)^2} \tag{4.29}$$

Finally, to evaluate performance of SocBIT in recovering ground truth *brand-conscious users*, we employ Area Under ROC curve (AUC) since the metrics can handle imbalance classes in our synthetic data when the ratio φ_b of brand-conscious users is less than 10% of the population.

For adoption prediction task, we compare our models against the following state-of-the-art recommendation models (i) Recommendation by Social Trust Ensemble a.k.a. *RSTE* [86], (ii) Social Recommendation a.k.a. *SoRec* [85]. We also use Non-negative Matrix Factorization a.k.a. *NMF* [71] as baseline.



(a) On training set

(b) On test set

Figure 4.5: Comparison of model RMSEs on synthetic training and test sets ($p_{train} = 80\%$)

4.4.3 Accuracy in Rating Prediction

First, we use cross validation to check if the models can figure out the ground truth number of topics. For that, we divide evenly the dataset into 5-folds, each of which occupies 20% of data. Each fold is then used as a test set while the remaining 80% of data is used as the corresponding training set. We then train the models by varying K in $[2,8]$. For each K value and each model, we compute the average $RMSE_{train}$ and $RMSE_{test}$ over five training and test sets respectively.

Figure 4.5a shows that, all model RMSEs are reduced when K increases and converge when $K \geq 5$. Moreover, Figure 4.5b shows that values $K > 5$ lead to overfitting. Thus, all models can discover that the ground truth number of topics is 5. The figures also demonstrate that SocBIT and SocBIT$^+$ not only fit better to training set but also outperform other models in adoption prediction task. In addition, SocBIT$^+$'s accuracy is just slightly lower than SocBIT.

Next, we examine how prediction accuracies of the models change when the amount of training data vary. The results are shown in Table 4.4a. As expected, with less amount of training data, the RMSEs get larger. However, SocBIT and SocBIT$^+$ still consistently outperform the other models regardless of the amount of data used for training.

4.4.4 Ground-truth Parameter Recovery

In this section, we first compare SocBIT$^+$ against SoRec, RSTE and NMF in terms of recovering ground truth topic factors θ_u's and θ_i's of users and items respectively. SocBIT is excluded from this comparison as it can return negative factors, which are not valid as topic factors. From Table 4.4b, we see that SocBIT$^+$ also outperform others in this task. Moreover, we also include errors $Err_{\mathbf{ub}}$ and $Err_{\mathbf{ib}}$ of SocBIT$^+$ in Table 4.4b. The errors are quite small, which shows that SocBIT$^+$ also performs well in recovering ground truth *brand factors* of users and items.

4.4.5 Brand-conscious Users Detection

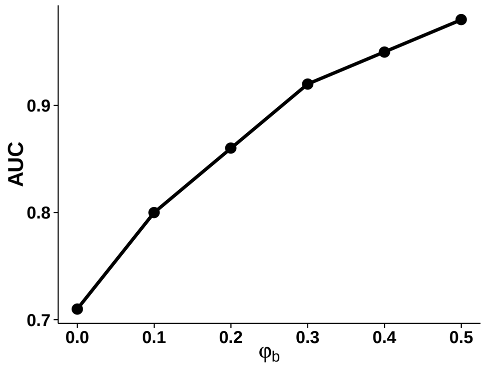


Figure 4.6: Performance of SocBIT$^+$ in brand-conscious user detection

Again, we exclude SocBIT from this task of detecting brand-conscious users as the model may return negative brand factors, which impedes the interpretation of the factors. We then consider the problem of detecting brand-conscious users as a binary classification task where users are classified as brand-conscious or not. Thus, we can use AUC measure to evaluate performance of SocBIT$^+$ in this task.

Table 4.4: Model comparison in (a) rating prediction, and (b) parameters recovery. All models are trained using 5 topics.

(a) Avg. $RMSE_{test}$ for various p_{train}

p_{train}	NMF	SoRec	RSTE	SocBIT	SocBIT$^+$
80%	0.22	0.17	0.16	**0.12**	0.13
60%	0.23	0.19	0.17	**0.14**	0.15
40%	0.30	0.24	0.23	**0.19**	0.20
20%	0.38	0.31	0.30	**0.27**	0.28

(b) Parameter recovery ($p_{train} = 80\%$)

Errors	NMF	SoRec	RSTE	SocBIT$^+$
$Err_{\mathbf{ut}}$	0.2	0.16	0.15	**0.13**
$Err_{\mathbf{it}}$	0.17	0.13	0.12	**0.11**
$Err_{\mathbf{ub}}$	NA	NA	NA	**0.14**
$Err_{\mathbf{ib}}$	NA	NA	NA	**0.20**

We generate different datasets embedded with ground truth brand-conscious users at different fraction φ_b of brand-conscious users, $\varphi_b \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ and train the two models on the generated datasets. After the training, we rate users by their decision preference values δ_u's, a user with (large) small δ_u is considered as (non) brand-conscious respectively. We then plot ROC curves and compute corresponding AUCs value obtained for each fraction φ_b.

From Figure 4.6, we can see that when $\varphi_b \geq 0.1$, SocBIT$^+$ has good performance with AUC values around 0.8. The more brand-conscious users exist in data, the more they improve in detecting them. When $\varphi_b \geq 0.3$, AUC values approach 1. This shows that SocBIT$^+$ can detect brand-conscious users extremely well when they occupy at least 30% of the user population.

4.5 Experiments on Real-world Data

In this section, we conduct experiments on two datasets obtained from *Foursquare* and *ACM Digital Library*. The first goal is to evaluate the performance of SocBIT and SocBIT$^+$ against other state-of-the-art methods in the task of adoption prediction. Secondly, we examine the learnt topic and brand factors from both SocBIT and SocBIT$^+$. Finally, we characterize the brand conscious users determined by the two models.

As later demonstrated in our experiments, the topics learnt by our models and other methods are similar. We thus place more emphasis on the evaluation of brand factors and brand-conscious users as these are novel contributions of our models.

4.5.1 Experimental Setup

For adoption prediction on the real-world datasets, we evaluate SocBIT and SocBIT$^+$ against RSTE, SoRec and NMF models. The evaluation metrics is RMSE. In all experiments, we set hyper-parameters $\lambda_{\mathbf{t}} = 0.001$, $\lambda_W = 1$, $\lambda_{\mathbf{b}} = 0.1$ and $\lambda_{\mathbf{d}} = 1$.

Given a dataset, we first used 5-fold cross validation (CV) to determine an appropriate number K of topics for each model. For each user with at least 5 adoptions, we divided his adoptions evenly into 5 folds. We iteratively use each fold as a test set and the others as the training set. For those users with less than 5 adoptions, we put all of his adoptions into the training set. We then trained each model using different *K*'s in $[5, 15]$ and determined the best *K* based on the average RMSE over the different test folds. This value of *K* will be fixed across all the models in later experiments. We also performed manual analysis on topics learnt by SocBIT$^+$ and provided them in Section 4.5.4.

Secondly, we examined the models accuracy in adoption prediction in Section 4.5.2. We looked at not only the change in accuracy w.r.t. the percentage of training data but also prediction accuracy of the models when they are applied on users with few observed adoptions. The latter is to see how our models fare against others in *cold-start* scenario.

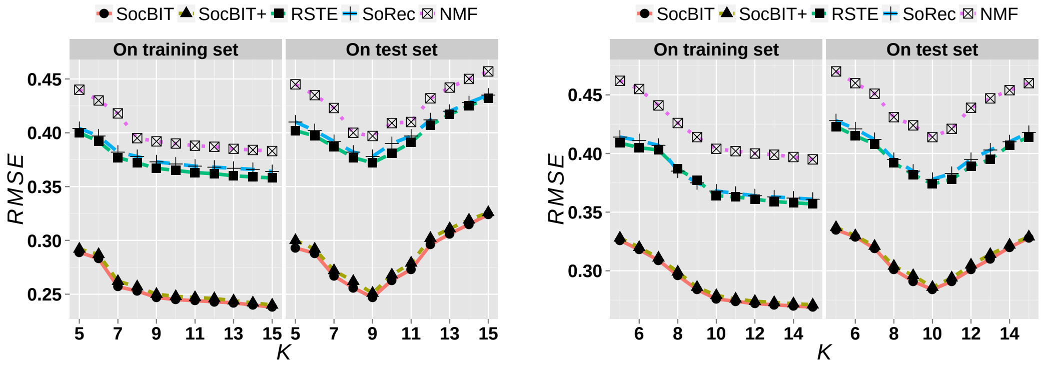
Thirdly, we analyzed brand-conscious users learnt by our models for both datasets in Section 4.5.3. We validated the results on brand-conscious users by showing that the brands they adopt either have high price (for **4SQDB** dataset) or high h-Index (for **ACMDB** dataset).

4.5.2 Evaluation on Adoption Prediction Task

First, we determine the appropriate number *K* of topics for each dataset by varying *K* from 5 to 15 and performed 5-fold Cross Validation (CV). The results are shown in Figures 4.7a and 4.7b for **4SQDB** and **ACMDB** respectively. From the figures, we can see that for **4SQDB**, the best value of *K* is 9, while for **ACMDB**, the best *K* is 10. These values of *K* will be used across all the models in later investigations.

Accuracy for Different Training Sizes

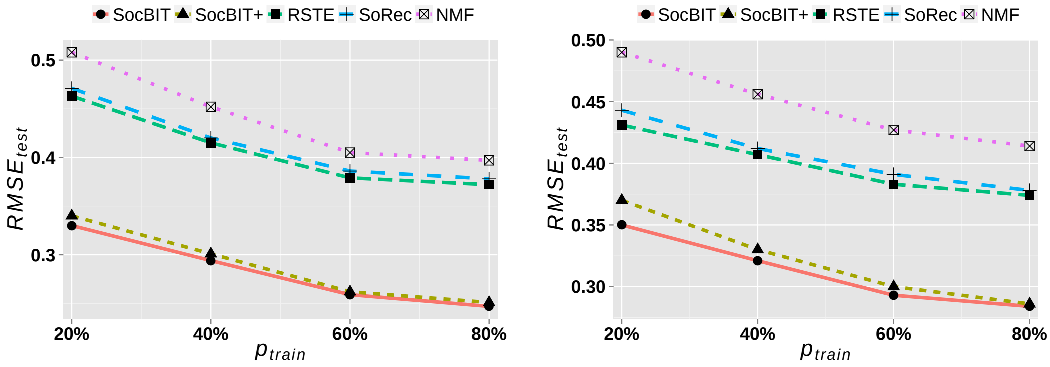
We train the models on the training set which occupies different percentage p_{train}'s of all adoption data. We then apply the trained models on the remaining adoption data. We use $p_{train} \in \{20, 40, 60, 80\}\%$ in the experiment. For example, $p_{train} = 80\%$ means we sample from each user 80% of his ratings to use for training and



(a) **4SQDB**

(b) **ACMDB**

Figure 4.7: Model RMSEs with respect to different K's ($p_{train} = 80\%$)



(a) **4SQDB** ($K = 9$)

(b) **ACMDB** ($K = 10$)

Figure 4.8: Model test RMSEs w.r.t. different training size

hold out 20% for testing.

Figure 4.8 shows accuracy of the models in adoption prediction using 9 topics for **4SQDB** and 10 topics for **ACMDB**. We observe that SocBIT and SocBIT$^+$ consistently outperform other models on both datasets over different training data sizes.

Accuracy for Different User Groups

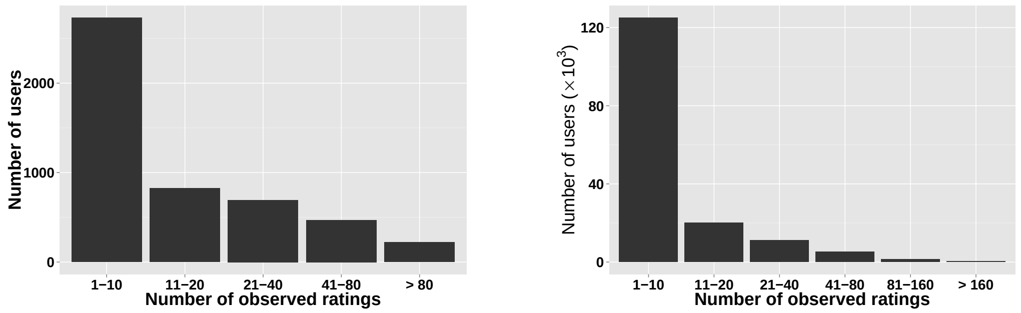
One of the challenges in recommendation systems research is to predict accurate ratings for a user even when she only rates a few items, i.e., *cold start* problem. We therefore want to investigate how well our model handles this problem. For that purpose, we first group users based on the number of observed ratings in training data, and then evaluate prediction accuracies for different groups. These groups have bin intervals covering different rating count ranges. The bins are “1 – 10”, “11 – 20”, “21 – 40”, “41 – 80” and “> 80” for **4SQDB**; and “1 – 10”, “11 – 20”, “21 – 40”, “41 – 80”, “81 – 160” and “> 160” for **ACMDB**. The count of each bin is shown in Figures 4.9a and 4.9b for **4SQDB** and **ACMDB** respectively.

From Figure 4.10, we observe that SocBIT and SocBIT$^+$ perform equivalently well and they both outperform other methods. Especially for users whose only few ratings can be observed, i.e. 1 – 10 ratings, our models offer much better prediction accuracy. The accuracy improvements of SocBIT$^+$ for this user group are as follows.

- On **4SQDB**: Compared with RSTE, SoRec and NMF, SocBIT$^+$ reduces RMSE by 32.3%, 33.2% and 36.5% respectively.
- On **ACMDB**: Compared with RSTE, SoRec and NMF, SocBIT$^+$ reduces RMSE by 22.8%, 24.2%, and 30.8% respectively.

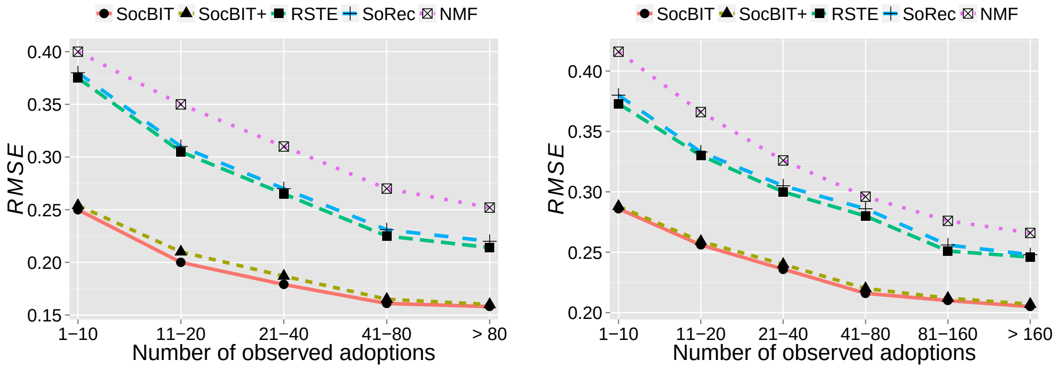
4.5.3 Brand-conscious User Identification

We now examine the brand-conscious users learned by SocBIT$^+$. Recall from Equation (4.3) that $1-\delta_u$ represents inverse of topic dependency weight of user u. Thus, for each user u, $1-\delta_u$ is a proxy for u’s *brand-consciousness* level. We then look at the distribution of different brand-consciousness levels inferred by SocBIT$^+$ on both datasets. Interestingly, both distributions follow a heavy tail form (see Figures 4.11a and 4.12a) with most users ($\approx$ 80%) having low brand-consciousness ($\leq$ 0.2) and



(a) **4SQDB** (b) **ACMDB**

Figure 4.9: User count of different rating count ranges



(a) **4SQDB** ($K = 9$)

(b) **ACMDB** ($K = 10$)

Figure 4.10: Model test RMSEs for user groups with different rating count ranges ($p_{train} = 80\%$)

only a very small proportion ($\approx 1\%$) of users having high brand-consciousness (≥ 0.8).

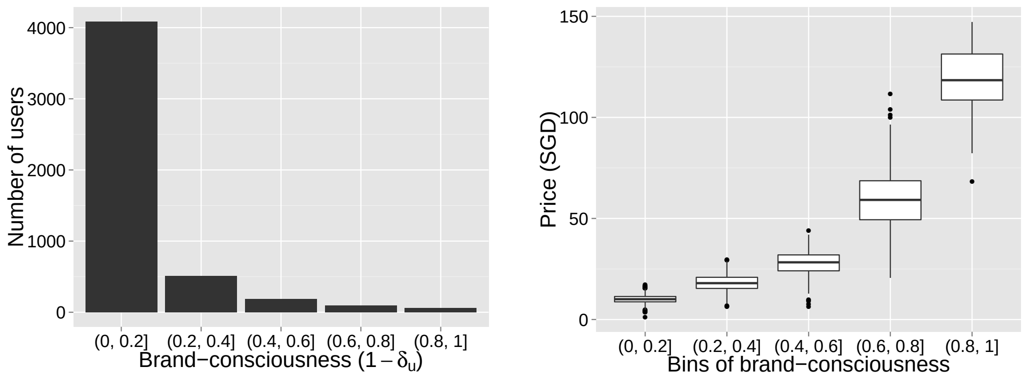
To further confirm the validity of this brand-consciousness measure, we check its relationship with empirical measures. Precisely, we check if the more brand-conscious users are, the more likely they adopt items from brand names. For this purpose, we need an empirical measure allowing us to determine brand names. For **4SQDB**, we use *brand price* to determine brand names. For **ACMDB**, we use *h-Index* to determine brand-name authors.

On 4SQDB

We collect venue prices from a food review website *hungrygowhere.com* and obtain prices of about 80% of the number of venues in **4SQDB**. We then estimate the price of each brand as the average price of venues of the brand.

We divide users by their brand-consciousness into 5 bins, namely (0,0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8] and (0.8, 1]. For each bin, we plot its price distribution of the brands adopted by users in the bin. The resultant box plot given in Figure 4.11b shows that the more brand-conscious users are, the more expensive brands they adopt. This matches our intuition.

Finally, we zoom in on the set of users who are highly brand-conscious, i.e., those with brand-consciousness more than or equal to 0.8. We denote the user set



(a) Distribution of user brand-consciousness

(b) Prices of brands adopted by users with different brand-consciousness

Figure 4.11: Analysis of brand-conscious users identified in **4SQDB**

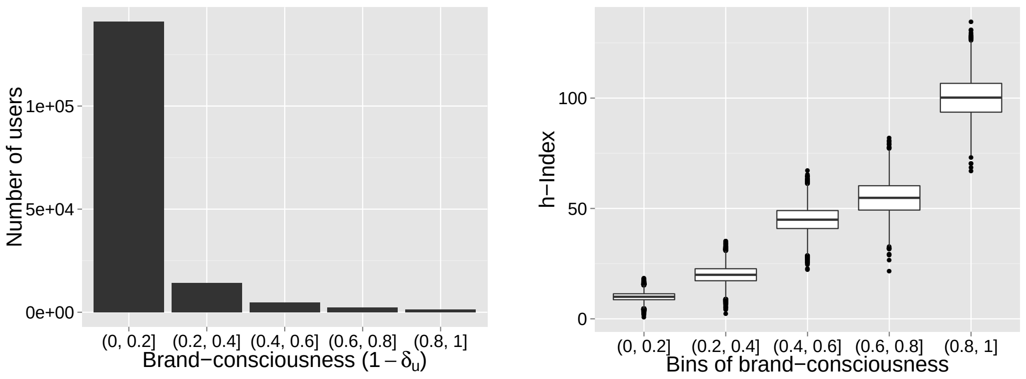
by *BCU* and compare them against normal users by the prices of the top-5 brands (in terms of adoption count) adopted by each user group. As shown in Table 4.5, users in *BCU* indeed adopt dining venues which are much more expensive than those adopted by normal user. The average price of brands adopted by the former is 163 (SGD) while that of brands adopted by normal users is just 8.4 (SGD). Moreover, all the top five brands adopted by users in *BCU*, are highly prestigious restaurants in luxury hotels or casinos in Singapore. These results match the intuition that highly brand-conscious users usually adopt expensive and/or prestigious brands. This again confirms that SocBIT$^+$ can discover brand-conscious users in a reasonable manner.

On ACMDB

We proceed similarly as the experiment with **4SQDB** by first dividing users into 5 bins of brand-consciousness levels (0,0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8] and (0.8, 1]. We then crawl h-Index of brands, i.e., authors, from Google Scholar. To

Table 4.5: Top-5 brands adopted by brand-consciousness users vs. those by normal users, the brands are sorted descendingly by adoption count. All prices are in SGD.

Top-5 brands adopted by users in *BCU*	**Price**	**Top-5 brands adopted by normal users**	**Price**
Punjab Grill	142	McDonald's	7
Pontini	142	Starbucks	10
Jaan	177	Swee Choon Tim Sum	13
Kaiseki Yoshiyuki	258	The Roti Prata House	6
Shinji by Kanesaka	335	Udders	6



(a) Distribution of user brand-consciousness

(b) h-Indices of brands adopted by users with different brand-consciousness

Figure 4.12: Analysis of brand-conscious users identified in **ACMDB**

overcome the limit on the number of authors we can crawl, we resort to using a sample for each bin. We randomly sample 200 authors adopted by users in the bin and collect the h-Index of those brands from Google Scholar. We then plot the distribution of those h-Indices. The results are shown in Figure 4.12b. We observe that more brand-conscious users adopt higher h-Index authors. This is similar to our observation on **4SQDB**.

4.5.4 Topics learnt by SocBIT$^+$ and NMF

This section shows topics learnt by SocBIT$^+$ and NMF for **4SQDB** and **ACMDB** datasets. On both datasets, the two models agree on the set of topics and differ only by a permutation. For easy reading, we reorder the topics learnt by NMF to align with those learnt by SocBIT$^+$. On **4SQDB**, we identify each topic by analyzing the top-10 restaurants under that topic. Each topic and the keywords in its top-10 restaurant names are provided in Table 4.6. The nine topics are the popular types of cuisines in Singapore; namely {Chinese, Indian, Japanese, Thai, American, Italian} cuisines, Seafood, Breakfast and BBQ. Similarly, we identified the ten topics in **ACMDB** based on title keywords of the top-10 papers of each topic. These topics and their keywords are provided in Table 4.7. The ten topics in **ACMDB** are the following ten research areas in Computer Science {Database, Data Mining, Software Engineering, System, Wireless/Sensor Network, Distributed

Table 4.6: Learnt topics for **4SQDB** by SocBIT$^+$ and NMF. The topics from NMF are re-ordered to align with those from SocBIT$^+$.

Topic (cuisine)	**Key words/phrases in top-10 venues**	
	SocBIT$^+$	NMF
American	steak, fast food, hamburger, Starbucks, tavern, French fries	pancakes, Starbucks, Astons Express, Swensen's, fast food, drive-in
Chinese	Beijing roasted duck, chicken rice, dim sum, mian (i.e. noodle)	porridge, yang chow fried rice, dim sum, si chuan food
Indian	roti prata, Indian food, lamb curry, curry, punjabi chicken	roti prata, curry, makhani chicken, shrimp curry, potato curry
Italian	Prego, Saizeriya, pasta, pizza, Oso Ristorante	PastaMania, pasta, Prego, pizza, Basilico
Japanese	sushi, sashimi, ramen, udon, Pepper Lunch Express	ramen, sushi, tempura, teriyaki, My Izakaya
Thai	green curry, pineapple fried rice, Pad Thai, jasmine rice	papaya salad, basil rice, red curry, green curry, Pad Thai
Seafood	seafood, chili crab, shrimp, fish-head steamboat, lobster	Korean seafood, fish-head steamboat, shark fin, chili crab, shrimp
Breakfast	pancakes, porridge, toast bread, half-boiled egg	coffee, toast, pancakes, fruit salad, bun
BBQ	BBQ, grill, Thai BBQ Korean BBQ	BBQ, Thai BBQ, BBQ buffet, Korean BBQ, outdoor BBQ

Systems, Security, IR, Internet, Machine Learning}.

4.6 Discussion

In this work, we have demonstrated that mining brand-related factors, e.g. user *brand consciousness* and user *brand preference*, can provide actionable insights to recommendation tasks. The insights can be leveraged to make more accurate rating prediction. Moreover, we propose two novel probabilistic matrix factorization models, namely SocBIT and SocBIT$^+$, to incorporate such brand factors and social network information. Our experiments on both synthetic and real-world datasets show that SocBIT and SocBIT$^+$ achieve the following.

- On synthetic data, both models outperform state-of-the-art models RSTE [86] and SoRec [85] in terms of rating prediction accuracy. Moreover, SocBIT$^+$ have good performance in recovering ground-truth factors as well as brand-conscious users embedded in synthetic data.

- More importantly, on real-world datasets, our models perform equivalently well and improve significantly accuracy of adoption prediction over RSTE and SoRec. Especially, for users with few observed ratings (1-10 ratings), both models offer much better accuracy. On **ACMDB**, they achieve improvements of 22.8% and 24.2% over RSTE and SoRec respectively. On **4SQDB**, they achieve improvements of 32.3% and 33.2% respectively.

- Although SocBIT performs slightly better in adoption prediction task, its gradient-based inference may return negative user and item factors, reducing interpretability. SocBIT$^+$ resolves this using a multiplicative inference, guaranteeing the non-negativity of learnt factors. The resultant factors learnt by SocBIT$^+$ are interpretable as topic and brand factors of users and items.

- SocBIT$^+$ infers user brand-consciousness from adoption data. The inferred brand consciousness scores follow a heavy tail distribution. Moreover, the more brand-conscious a user is, the more likely he adopts items from prestigious brands, characterized by expensive price on **4SQDB** or large h-index on **ACMDB**.

In this work, we manually tune regularization coefficients λs. In our future work, we want to develop an automatic tuning method. We also plan to combine user brand-consciousness and brand preference to infer whether a given brand is exclusive. Finally, when users are dependent on brand in item adoption, we can develop new methods to profile their attributes based on information of the brands they adopt.

Table 4.7: Learnt topics for **ACMDB** by SocBIT$^+$ and NMF. The topics from NMF are re-ordered to align with those from SocBIT$^+$.

Topic	**Key phrases in top-10 papers**	
	SocBIT$^+$	NMF
Database	large DB, relational DB, key, joint operations, query, aggregation, (semi)-structured DB	key, foreign key, DB, query, XML, (semi)-structured DB, structural join
Data Mining	mining, data, clustering, frequent patterns, k-means, classification, (un)supervised, SVM	classify, regression, SVM, k-means, text mining, (un)supervised, association rule
Software Engineering	Java, C++, development, path profiling, function calls, compiler, programming	garbage collection, C, dependence graph, compile, software, procedure, bug localize
Internet	network, lookup service, protocol, WWW, peer-to-peer, Internet, packet dynamics	network, IP traceback, latency, WWW, Internet, protocol, congestion, traffic
System	file system, performance, system design, caching, OS, (multi)processor, cache	system, caching, architecture, cache, deadlock, processor, TinyOS, battery
Wireless Network	wireless, sensor networks, routing, directed diffusion, protocol	routability, placement, accurate, distributed sensor networks, router
Distributed Systems	race detector, failure detectors, order, distributed systems, clocks, lock-free	deadlock, workload, race, distributed consensus, multithreaded, parallel, schedule
Security	wireless security, cryptography, public-key, symmetric-key, digital signatures, anonymity	privacy, protect, access control, anonymity, anonimous, public-key
Information Retrieval	index, inverted index, query, text/image retrieval, distributed IR	inverted index, text retrieval/categorization, relational algebra, IR, query
Machine Learning	online learning, neural network, Bayes, image recognition, intelligent agents	batch/transfer/statistical learning, (un)supervised, machine translation, Bayes

Chapter 5

A Diffusion Model with Item Interaction and Homophily

5.1 Introduction

While many items may diffuse simultaneously in a social network, most existing models of diffusion are built upon *independent* contagion assumption whereby the diffusion of each item is assumed (at least implicitly) to happen independent of other items. The interaction among items during diffusion is thus left out of the picture. This is obviously not true in the complex dynamics of diffusion process. For instance, the diffusion of iPhones in the Facebook friendship network may interact favorably with that of iPad; and the diffusion of a catchy phrase on Twitter also aids the diffusion of its variants.

Interaction among items. Modeling these interactions is crucial in both theory and practice since it helps us understand the detailed dynamics of multiple item diffusion. With a diffusion model that incorporates item interaction, businesses will be able to develop suitable strategies to promote diffusion of their own items considering the other items that have been diffused recently or are being diffused. It may be good to time the diffusion of a new item with the diffusion of other similar items (possibly by the business or other businesses) to achieve a larger reach. This idea of diffusion with item interaction can be further illustrated in the following motivating example.

Example. *A user may be inspired to watch the movie version of "Hunger Games" after observing some neighbors already read the book. Moreover, if both the book and the movie versions were adopted by a neighbor, the user will even be more likely to adopt the movie than if only one of them was adopted by the neighbor (as he may be more convinced that the movie is good in the former case).*

The example not only highlights that diffusion of an item can support that of another *similar* item but suggests other deeper ideas which will distinguish our work from the rest. These ideas are:

1. *The more similar items are, the more interaction will happen between them in diffusion.* In other words, item similarity can be used as a proxy for *item interaction*. This idea will be formulated in Section 5.2.2 where we propose a general diffusion framework for modeling item interaction when there is more than one item diffusing.

2. *Whether or not a user adopts an item i is affected not only by neighbors who have adopted exactly the item but also by those who have adopted other items.* A neighbor who has already adopted another item i' can still influence the decision when i' is very similar to i.

3. *Each neighbor's social influence on a user's adoption decision should include all contributions from a set of items adopted by the neighbors*, not just limited to one item as in the existing models.

The work in [134] explores a somewhat similar scenario, namely *bundle* diffusion. The authors also propose the idea that items adopted by peers can affect adoptions of a given user. However, there are major differences between our work and theirs as follows.

- Their work considers diffusion of a whole bundle of items, not diffusion of each individual item. Meanwhile, our work studies multiple cascades, each of which is for one single item.

- In our work, the items may support each other to diffuse and this support effect is explicitly modelled. This is not the case in [134].

Impact of Homophily. Another important aspect which also has great impact on item diffusion, is the well-known *homophily* phenomenon. Homophily refers to the tendency of individuals to associate and bond with similar others. It is well known that homophily affects the mechanisms in which item diffusion happens, be it innovation [108], information [27] or behavior [20]. Thus, it is important to integrate homophily into diffusion models so that we can better quantify its effect on diffusion. In this work, we assume a global homophily level of the network and learn it from the diffusion cascade data. Given that networks with homophily involves more similar users connecting with one another, it also plays a role in determining if an item can more smoothly diffuse to across the network links.

5.1.1 Research Problem and Contributions

In this chapter, we therefore propose to consider both item interaction and homophily in the design of a new microscopic diffusion model. Specifically, our research problem is as follows.

Problem. *Given a set of items I, a social network $G = (V, E)$ where there exists homophily effect and history of adoption of items in I, we aim to develop a microscopic diffusion model which can capture the homophily effect and item interaction during diffusion processes.*

To involve both *item similarity* (as proxy of item interaction) and *user similarity* (due to homophily), our modeling approach employs latent factors (LF) to represent both items and users (e.g. [96], [68]) where each user or item is represented as a vector in a common feature space with dimension much smaller than the number of items and users. The similarity between two items (or users) can then be defined by the cosine similarity of the respective item (or user) vectors. Unlike the collaborative filtering approach taken by recommender systems, our diffusion modeling work also consider social influence among users. Although there are recently hybrid models ([67], [85]) which combine latent factor approach with social networks, they still do not model a user adopting an item due to influence by the neighbors' past adoption of *similar* items and the strength of relationships with these neighbors.

Summary of contributions. We make the following contributions.

- We develop a diffusion framework which incorporates both item interaction and homophily into modeling diffusion. To the best of our knowledge, this is the first attempt to combine the two factors. The framework is flexible and can offer useful insights to multiple item diffusion.

- We propose a specific diffusion model based upon the new framework. This model known as TIHAD utilizes latent factors to capture item interaction and neighbor similarity due to homophily to effectively model diffusion processes of multiple items.

- We formulate the parameter learning of model as a constrained optimization problem, and devise an effective learning algorithm using Projected Gradient Descent technique.

- We conduct experiments on both synthetic and real datasets to show that: (a) homophily increases diffusion significantly, and (b) item interaction at topic level boosts diffusion among similar items. We also shows that TIHAD outperforms the baseline model in the hashtag adoption prediction task.

5.1.2 Related Work

In the following, we provide a brief review of works related to this chapter.

Ecology-based Diffusion Models. A possible approach to modeling interacting diffusion processes comes from Dynamical Systems theory for diffusion of species.

In this approach, diffusion processes are modeled as solutions of some system of partial differential equations, e.g. Lotka-Voltera system for predator-prey [135]. However, such systems get complicated very fast and solving becomes so difficult or even impossible. In fact, it has been shown that even when there are only three species in the system, the dynamics of the system is already very complicated with peculiarities [92]. Thus, most works by this approach [63, 102] are limited to the case of two items.

Social Influence and Diffusion Models. Social influence modeling works take into account social interest and social trust as additional input to achieve better accuracy for recommendation [28, 67, 85, 115, 121]. These works proposed various ways of modeling the social dimension such as factorizing the social network graph [85] or modeling social factors of users as another set of latent factors ([115], [28]). While these works focus on recommendation tasks, they are similar to diffusion models in that both estimate social influence on user-item adoptions. Social diffusion models on the other hand consider only influence from a subset of neighbors, called the set of *active* neighbors $\mathscr{A}_u$, who adopt exactly the target item ([45, 62, 76]). For example, Linear Threshold (LT) model is a social diffusion model which estimates social influence by the sum of weights of active neighbors only i.e. $social\ influence = \sum_{v \in \mathscr{A}_u} w_{v,u}$. As pointed out in our motivating example, items similar to the item being diffused i can affect its diffusion. Hence, even though a neighbor has not yet adopted item i, he can still affect the target user's decision on adopting i, when the neighbor adopted item(s) similar to i. Such a diffusion scenario has been largely overlooked in the existing social diffusion models.

Chapter Outline. In Section 5.2, we then present our general framework and a derived diffusion model known as TIHAD. The learning of this model is given in Section 5.3. Section 5.4 describes experiments that evaluate the TIHAD using both synthetic and real datasets. We finally conclude the chapter in Section 5.5.

5.2 Framework and Models

Before we present our proposed modeling framework and the TIHAD model, we first introduce the notations used in the problem formulation.

5.2.1 Basic Notations

We continue to use notations θ_u and θ_i to denote latent factors of a user u and an item i respectively. We represent the social network as a (directed), weighted graph $G = (V,E)$ whose nodes represent users and edges represent links among the users. For each edge (u,v), the edge weight $w_{v,u}$ represents the social influence that v exerts on u. To model diffusion over the network during a time period, we bin the continuous time into discrete time steps $\{1,2,\cdots,T\}$ and consider adoptions in each step.

Unlike the latent factor models which focus on user-item interactions only, our work considers both user-item and item-item interactions in the diffusion setting. We are therefore also interested in the effect of item similarity. Naturally, we can estimate the similarity between two items i and j by the inner product $\theta_i^T \theta_j$ in the latent factor space. In this chapter we also follow the common practice of considering only positive latent factor vectors [96, 111] for interpretability.

Denote adoption decision of a user u on item i at time step t as $a_{u,i,t}$. At first sight, it seems that $a_{u,i,t}$ is simply a binary label which is 1 when u adopt i and 0 otherwise. However, it is often that a user does not adopt an item because he has not been exposed to the item. It is thus incorrect to assume that he rejects the item, and underestimate his preference for the item. We can avoid this by considering, at each time step, only items which are exposed to the user. When the user did not adopt an item he has exposed to, we say that the case is a non-adoption. We call these user-exposed items as the candidate items in Definition 12, which in turn help us to define adoption labels properly in Definition 13.

Definition 12 (Candidate item)**.** *At a given time step t, a candidate item for a user is* *an item that: (i) he has not yet adopted before t; and (ii) he is exposed to it through some source (e.g. through his neighbors). The set of candidate items for a user u at time t is denoted by $C_{u,t}$.*

Definition 13 (Adoption label)**.** *Given an item $i \in C_{u,t}$, adoption label $a_{u,i,t}$ is a binary variable which is* 1 *if u adopts i at time t and* 0 *otherwise.*

5.2.2 Framework

Our proposed framework extends the latent factor model framework by considering both personal interest and social influence in the modeling of user-item adoption at different time steps. Personal interest is estimated by user-item similarity in a latent space and social influence is an aggregation of individual influences from neighbors. However, that influence from a neighbor v now depends on: (i) the link weight $w_{v,u}$, and (ii) the interaction level between item i and a certain set of items adopted by v. We also follow common practice (e.g. [67]) by including in the framework global bias μ, user bias b_u and item bias b_i.

We first state the core formula of the framework in Equation 5.1 and provide the reasoning behind the formulae subsequently. By denoting personal interest and social influence as $\phi(u,i)$ and $\sigma(u,i,t)$ respectively, we can express the framework as follows.

$$\widehat{a}_{u,i,t} := \mu + b_u + b_i + \overbrace{\theta_u^T \theta_i}^{\phi(u,i)} + \overbrace{\sum_{v \in N_u} w_{v,u} \cdot \lambda\,(v,t,i)}^{\sigma(u,i,t)} \tag{5.1}$$

where

1. $w_{v,u}$: link weight, which will later be estimated by a function of user similarity parameterized by *homophily level h*

2. $\lambda\,(v,t,i)$: the *interaction level* between the items adopted by v and item i at time step t.

Our framework adapts the general formula by proposing in Equation 5.1 a novel estimation of social influence term $\sigma(u,i,t)$ and a homophily derived link weight $w_{v,u}$. As can be seen from the definitions, the estimation will incorporate both *item interaction* and *homophily factor*. To keep the framework tractable, we assume the latent factors are static. Given this framework, we can now apply it for modeling *interacting diffusion* processes of items over a social network as follows.

Framework (Interacting Diffusion of Items)**.** *Consider a set of items I and a social network G. For each such candidate item i, its adoption label $\widehat{a}_{u,i,t}$ can be estimated by Equation 5.1. Candidate i will be adopted by u if the estimation is close enough to* 1*. Thus, at each time step, a user can adopt several candidate items which satisfy this criterion. The process continues until no more adoption can happen.*

We proceed by providing the logic behind Equation 5.1 of our framework. The logic includes two parts: how to define item interaction and how to incorporate homophily.

Item interaction

The interaction level depends on a certain set of v's adopted items which can actually affect u's decision. This leads us to the concept of *effective item set* defined as follows.

Definition 14 (Effective item set)**.** *For a given neighbor v of user u, the set of items adopted by v which can influence adoption decision $a_{u,i,t}$ is called* effective item set *from the neighbor at time step t and denoted as $I_{eff}(v,t)$.*

Given effective item set $I_{eff}(v,t)$, we now need to estimate the interaction level $\lambda\,(v,t,i)$ between the adopted items of v and candidate item i and time step t. We now provide a general estimation of $\lambda\,(v,t,i)$ in Definition 15.

Definition 15 (Interaction level)**.** *The interaction level $\lambda(v,t,i)$ is defined as the sum of interactions (i.e. similarities) between the effective item set of v and i at time step*

t.

$$\lambda(v,t,i) := \sum_{j \in I_{eff}(v,t)} \theta_j^T \theta_i \tag{5.2}$$

The social influence from neighbor v will then be $w_{v,u} \times \lambda(v,t,i)$. In total, social influence on u will be estimated by

$$\sigma(u,i,t) := \sum_{v \in N_u} w_{v,u} \times \lambda(v,t,i) = \left(\sum_{v \in N_u} \sum_{j \in I_{eff}(v,t)} w_{v,u} \theta_j \right)^T \theta_i \tag{5.3}$$

Note that for directed networks, N_u will be replaced by the followee set of u.

Replace 5.3 into 5.1, we obtain our novel estimation for adoption label

$$\widehat{a}_{u,i,t} := \mu + b_u + b_i + \theta_u^T \theta_i + \left(\sum_{v \in N_u} \sum_{j \in I_{eff}(v,t)} w_{v,u} \theta_j \right)^T \theta_i \tag{5.4}$$

This new estimation allows our framework to capture item interaction. Thus, in the context of interacting diffusion, we expect it to provide a better model than existing models ([62, 76]). This will be realized later in our experiments on synthetic data.

Incorporating homophily

Equation 5.4 involves link weight $w_{v,u}$ which is determined by homophily factor. Due to homophily effect, more similar individuals tend to be connected. We therefore propose to estimate $w_{v,u}$ as an *increasing* function of the similarity between u and v. In other words, for a social network with an underlying homophily level $h \in [0,1]$ (smaller h implies low homophily), we propose to define $w_{v,u}$ as:

$$w_{v,u} := g(\theta_u^T \theta_v | h) \tag{5.5}$$

where $g(.)$ is an increasing function parameterized by h. Since weights are in $[0,1]$, we also choose functions g with range in $[0,1]$.

Finally, by replacing Equation 5.5 in 5.4 and using estimation of $\lambda(v,t,i)$, we obtain Equation 5.6, the main estimation of our framework.

$$\widehat{a}_{u,i,t} := \mu + b_u + b_i + \theta_u^T \theta_i + \sum_{v \in N_u} g(\theta_u^T \theta_v | h) \times \sum_{j \in I_{eff}(v,t)} \theta_j^T \theta_i \tag{5.6}$$

5.2.3 Proposed Model

To apply our general framework, we need to give specific definitions for $g(\theta_u^T \theta_v | h)$, and $I_{eff}(v,t)$. This leads to our proposed Topic Interaction and Homophily Aware Diffusion (TIHAD) Model.

In TIHAD, we define the function $g(.)$ as a linear function of user similarity $\theta_u^T \theta_v$ as follow.

$$g(\theta_u^T \theta_v | h) := h \times (\theta_u^T \theta_v),\ \forall (u,v),\ v \in N_u \tag{5.7}$$

There are other interesting forms of function $g(.)$ including $\left(\theta_u^T \theta_v\right)^h$. In this work, we focus on the linear form due to its tractability and leave other forms for future research.

For $I_{eff}(v,t)$, we choose the set of items *adopted recently* by neighbor v. This is based on the common intuition that a user usually pays attention only to those recent items (e.g., Twitter users only focus on recent hashtags from their followees [129]). Thus, for each time t, we choose the effective set as the set of k items which neighbor v adopted most recently with respect to time step t, which we denote as $r_v^{k,t}$. Hence, $I_{eff}(v,t) = r_v^{k,t}$.

The TIHAD model is therefore expressed as Equation 5.8.

$$\widehat{a}_{u,i,t}^{tihad} = \mu + b_u + b_i + \theta_u^T \theta_i + h \theta_u^T \left[S_t(u)\right] \theta_i \tag{5.8}$$

where the matrix $S_t(u)$ is

$$S_t(u) := \sum_{v \in N_u} \theta_v \ (\sum_{j \in r_v^{k,t}} \theta_j)^T \tag{5.9}$$

$S_t(u)$ can be interpreted as the matrix characterizing the social influence from u's neighbors recent adoption events.

5.2.4 Linear Threshold with Latent Factors (LTLF)

In the special case when $I_{eff}(v,t) = \{i\}$, Equation 5.3 becomes

$$\tilde{\sigma}(u,i,t) = \left(\sum_{v} w_{v,u}\right) ||\theta_i||^2$$

where v now is not an arbitrary neighbor of u but instead an *active* neighbor i.e. one who actually adopted i. This estimation of social influence is obviously an extension of Equation 5.3 commonly used in Linear Threshold models ([45], [19]). Thus, by substituting it into Equation 5.4, we obtain the following model, called Linear Threshold with Latent Factors (LTLF)

$$\widehat{a}^{ltlf}_{u,i,t} := \mu + b_u + b_i + \theta_u^T \theta_i + \left(\sum_{v \in A^i_{u,t}} w_{v,u}\right) ||\theta_i||^2 \tag{5.10}$$

5.3 Model Inference

We formulate the learning of TIHAD model parameters as a constrained optimization problem, which can be solved by Projected Gradient Descent (PGD). We provide the detailed formula to solve the problem and a pseudocode for model learning. For brevity, we use P and Q matrices to denote user and item latent factors respectively. All parameters of TIHAD then can be compactly represented by $\Pi = (h, \mu, \{b_u\}_{u \in U}, \{b_i\}_{i \in I}, P, Q)$. We also use $\widehat{a}_{u,i,t}$ in place of $\widehat{a}^{tihad}_{u,i,t}$ for brevity.

5.3.1 Optimization Formulation

Let A_1^T denote the set of all adoption labels in a diffusion cascade during the time span $[1,T]$.

$$A_1^T := \{a_{u,i,t} : t \in [1,T],\ u \in U \text{ and } i \in C_{u,t}\} \tag{5.11}$$

Diffusion data is then represented by a tuple of item set, the social network and the adoption labels as $\mathscr{D} = \left(I, G, A_1^T\right)$. Given $\mathscr{D}$, we formulate the model learning problem as finding the optimal parameters Π^* that minimize squared error upon generating the adoption labels.

For a given Π, the squared error at time step t is the sum

$$SE_t(\Pi|\mathscr{D}) = \sum_{u \in U} \sum_{i \in C_{u,t}} [\widehat{a}_{u,i,t}(\Pi) - a_{u,i,t}]^2 \tag{5.12}$$

Hence, over the whole time span $[1,T]$, the total error is

$$\mathscr{E}(\Pi|\mathscr{D}) = \sum_{t=1}^{T} SE_t(\Pi|\mathscr{D}) = \sum_{t=1}^{T} \sum_{u \in U} \sum_{i \in C_{u,t}} [\widehat{a}_{u,i,t}(\Pi) - a_{u,i,t}]^2 \tag{5.13}$$

To avoid over-fitting, we also define a regularizer as

$$\mathscr{R}(\Pi) := h^2 + \sum_u b_u^2 + \sum_i b_i^2 + \|P\|_F^2 + \|Q\|_F^2 \tag{5.14}$$

where $\|.\|_F$ denotes the usual Frobenius norm. Hence, the objective function is

$$J(\Pi|\mathscr{D}) = \frac{1}{2}\,[(\mathscr{E}(\Pi|\mathscr{D}) + \delta \mathscr{R}(\Pi)]$$

We now can formulate the learning as the following constrained optimization problem.

Problem. *Given diffusion data set* $\mathscr{D} = \left(I, G, A_1^T\right)$*. We learn parameters* Π *by solv-*

ing for optimal parameters which minimize the objective function

$$\Pi^* = \underset{\Pi}{\operatorname{argmin}} J(\Pi|\mathscr{D}) = \underset{\Pi}{\operatorname{argmin}} \frac{1}{2}\left[\mathscr{E}(\Pi|\mathscr{D}) + \delta\mathscr{R}(\Pi)\right] \tag{5.15}$$

subject to constraints

$$\theta_u \geq 0,\ \forall u \in U, \quad \theta_i \geq 0,\ \forall i \in I \quad \mathit{and}\ 0 \leq h \leq 1 \tag{5.16}$$

5.3.2 Optimization Solution

In general, the above problem is not convex. Thus, we resort to a solver which uses grid search and Projected Gradient Descent (PGD). For that, we provide formulae of gradients in the following sections. Due to space constraints, proofs of these formulae are provided in Appendix B.

Derivatives for bias variables

$$\frac{\partial J}{\partial \mu} = \sum_t \sum_{u \in U} \sum_{i \in C_{u,t}} \overbrace{\left(\widehat{a}_{u,i,t}(\Pi) - a_{u,i,t}\right)}^{e_{u,i,t}} \tag{5.17a}$$

$$\forall u \in U,\ \frac{\partial J}{\partial b_u} = \delta b_u + \sum_t \sum_{i \in C_{u,t}} e_{u,i,t} \tag{5.17b}$$

$$\forall i \in I,\ \frac{\partial J}{\partial b_i} = \delta b_i + \sum_t \sum_{u \in U : i \in C_{u,t}} e_{u,i,t} \tag{5.17c}$$

Derivative for homophily variable

$$\frac{\partial J}{\partial h} = \delta h + \sum_t \sum_u \theta_u^T \left[S_t(u)\right] q_t^{err}(u) \tag{5.18}$$

where $S_t(u)$ is defined in Equation 5.9 and $q_t^{err}(u) := \sum_{i \in C_{u,t}} e_{u,i,t} \cdot \theta_i$.

Derivatives for user and item factors

1. (Gradient w.r.t θ_u) For each given user u, we have

$$\nabla_{\theta_u} J = \delta\theta_u + \sum_t \left[M_t(u) q_t^{err}(u) + h\eta_t(u) q_t^k(u)\right] \tag{5.19}$$

Algorithm 3 PGD for TIHAD model using an initial guess Π_0

```
procedure Train($\mathscr{D}$, $\Pi_0$, $\varepsilon$)
    Initialize $\Pi_c \leftarrow \Pi_0$
    while $(!converge)$ do
        Compute objective value: $j_c \leftarrow J(\Pi_c|\mathscr{D})$        ▷ use Eqns. 5.13 – 5.15
        Compute gradients: $g_c \leftarrow \nabla J(\Pi_c|\mathscr{D})$        ▷ use Eqns. 5.17a – 5.21
        Descend: $\Pi_n \leftarrow$ GRADPROJ($\Pi_c, j_c, g_c$)    ▷ see gradproj() in [61]
        Check convergence: $converge \leftarrow (\ |\Pi_n - \Pi_c| < \varepsilon\ )$
        $\Pi_c \leftarrow \Pi_n$
    end while
    return $\Pi_n$
end procedure
```

where matrix $M_t(u)$ and scalar $\eta_t(u)$ are defined as

$$M_t(u) := Id + hS_t(u) \text{ and } \eta_t(u) := \sum_{v \in N_u} \theta_v^T q_t^{err}(v) \tag{5.20}$$

where Id denotes the identity matrix.

2. (Gradient w.r.t. θ_i) For each given item i, we have

$$\nabla_{\theta_i} J = \delta\theta_i + \sum_t \left(h \sum_{u \in U} \left[q_t^{err}(u) \varphi_{u,i,t}^T \right] \theta_u + \sum_{u: C_{u,t} \ni i} e_{u,i,t} \left[M_t(u) \right]^T \theta_u \right) \tag{5.21}$$

where vector $\varphi_{u,i,t} := \sum_{recent\ adopters} \theta_v$ is the sum of factors of neighbors who adopted i recently.

Now that all derivatives are available, we can use them in Projected Gradient Descent (PGD) with grid search to update the corresponding parameters. Thus, we repeat Algorithm 3 with different initial parameter values to learn the parameters of TIHAD model. All the derivatives in the algorithm are computed using Eqns. 5.17a – 5.17c and 5.18 – 5.21.

5.4 Experiments

In this study, we want to be able to evaluate TIHAD model with some parameter settings that control the item interaction and homophily factor during the diffusion

Algorithm 4 Generation of a network with a given homophily level

```
procedure BuildNetwork($U$, $N_e$, $h$)
    $Pairs \leftarrow \{(u,v) : u \neq v \in U\}$
    for each user pair $(u,v) \in Pairs$ do
        Compute user-item similarity: $sim(u,v) \leftarrow \theta_u^T \theta_v$
        Compute edge weight: $\rho(u,v) \sim \exp(h \cdot sim(u,v))$
    end for
    Normalize: $p(u,v) \leftarrow \frac{\rho(u,v)}{\sum \rho(u',v')}$, $\forall (u,v) \in Pairs$
    Collect probabilities: $probs \leftarrow (p(u,v) : (u,v) \in Pairs)$
    Sample $N_e$ edges by the probabilities: $E_h \leftarrow sample(Pairs, N_e, probs)$
return Network $G_h = (U, E_h)$
end procedure
```

process. Hence, we need a synthetic diffusion data generation method with the following input parameters: (a) M items, (b) N users, (c) N_e relationships among the users, (d) f latent factors, (e) homophily value h for the social network, (f) T number of time steps, and (g) k recently adopted items. The generation steps are described below:

1. (Generation of M items and N users in latent space) We generate M items and N users as f-dimensional vectors θ_i's and θ_u's respectively. The item and user vectors are generated such that each of them has a dominant factor. The set of users and items are denoted by U and I respectively.

2. (Generation of a social network with homophily value h) We generate N_e edges among the users using Algorithm 4. The resultant network, $G_h = (U, E_h)$ where E_h denotes the set of N_e edges, satisfies the required homophily level h.

3. (Generation of an initial adoption state) We want to ensure that every user in the network initially has adopted at least k items. We assign k items to each user based on his latent factor interests.

4. (Generation of a diffusion cascade) We randomly assign a user as the single seed of diffusion. The seed user will adopt all M items initially. We then employ TIHAD model to start generating a data set of simultaneous diffusion

Algorithm 5 Generation of diffusion data

```
procedure CreateDiffusion($I$, $G_h$, $\theta$, $T$, $u_s$)          ▷ $G_h$: from Algo. 4
    for $t \in [1,T]$ do
        Initialize set of adoption records at time $t$: $A_t \leftarrow \emptyset$
        for $u \in U$ do
            findAdopts($u, t, u_s, \theta$)          ▷ find items $u$ will adopt at $t$
            $A_t \leftarrow A_t \cup \{(u,i,t) : i \in I_t(u)\}$          ▷ add to adoptions at $t$
        end for
    end for
    Collect all adoption records: $A_1^T \leftarrow \bigcup_{t=1}^{T} A_t$
    return $\mathscr{D} = \left(I, G_h, A_1^T\right)$
end procedure
function findAdopts($u, t, u_s, \theta$)
    Find $C_{u,t}$ by Def. 12          ▷ use seed $u_s$ to get $C_{u,1}$
    for candidate item $i \in C_{u,t}$ do
        Estimate adoption label $\widehat{a}_{u,i,t}$ by Equation 5.8
    end for
    Pick adoptions: $I_t(u) \leftarrow \{i \in C_{u,t} : \widehat{a}_{u,i,t} \geq 1 - \theta\}$
    return $I_t(u)$
end function
```

of the items over the network G_h within the time interval $[1,T]$. The details of this step are given in Algorithm 5.

We generate N diffusion cascades by performing steps 3 and 4 with a different initial adoption state and different user as the seed each time. Hence every diffusion cascade share the same network with identical user and item latent factor vectors. We finally generate N different data sets so that we can get empirical distribution of cascade sizes.

5.4.1 Impact of Homophily on Diffusion

Experiment Setup: We study how the size of diffusion cascade is affected by different degrees of homophily h. Thus, we generate items and users by setting $f = 10$. We then generate diffusion in five different networks G_h's each with a different h value, $h \in \{0, 0.2, 0.4, 0.6, 0.8\}$. These networks however share the same set of users and same number of edges to minimize the effect of choices of users and number of relationships among them. For each such network, we generate N diffusion cascades of M items using TIHAD and study distribution of the average cascade size

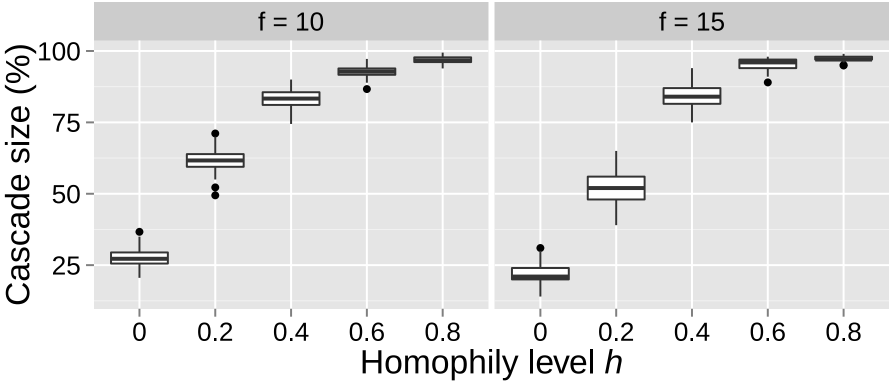


Figure 5.1: Impact of homophily on multi-item diffusion (cascades generated by TIHAD under different settings of number of factors f)

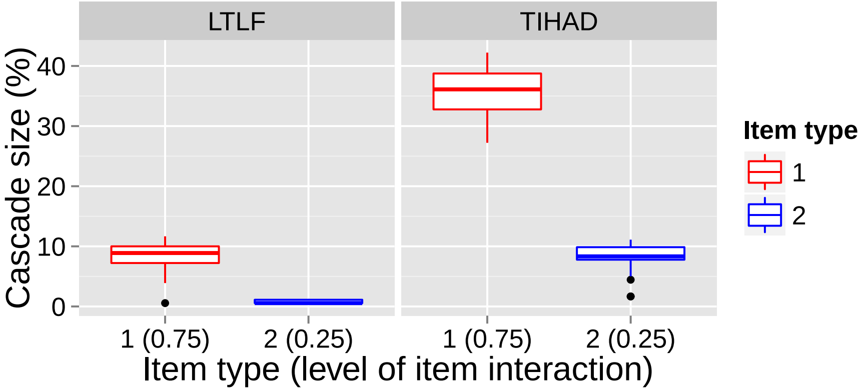


Figure 5.2: Impact of item interaction on multi-item diffusion (cascades generated by both models, for TIHAD we set parameters $h = 0.1$ and $f = 10$)

over the M items. Detailed statistics of this experiment is provided in Table 5.1.

Result: As the homophily level increases, the diffusion cascade also becomes larger (see Figure 5.1). This trend is observed for all items. To evaluate the robustness of the result, we repeat the experiment for $f = 15$ and $f = 20$. We report here results for $f = 10$ and $f = 15$. This result is expected as homophily facilitates diffusion ([43], [27]). It also shows that our model has incorporated homophily effect properly.

5.4.2 Impact of Item Interaction on Diffusion

Experiment Setup: In this experiment, we change our focus to study how item interaction (i.e. support among items) affects diffusion. We now generate diffusion cascades on the same network with a fixed homophily level $h = 0.1$. The item set

Table 5.1: Parameters used in synthetic data generation

# factors (f)	# items	# users	# edges	Homophily level (h)	# recent items (k)	# time steps (T)
$10,15,20$	100	500	$70K$	$0,0.2,0.4,$ $0.6,0.8$	5	20

Table 5.2: Statistics of diffusion data among Singapore Twitter users in Valentine Day

Dataset	# hashtags	# users	# follow links	# adoptions	# time steps	# labels
Training	4002	1000	9935	$11,565$	12	$60,875$
Test	1219	884	8754	9390	12	$39,375$
Total	4002	1000	9935	$20,955$	24	$100,250$

is however generated differently. We partition the item set I into the *majority* set I_1 (occupy 75% of I) and the *minority* set I_2. In each subset, items are generated such that they are similar to each other. Thus, items in I_1 receive more interaction than items in I_2 and we can study difference in cascade sizes of items in two sets. Other statistics of this experiment is the same as in Table 5.1.

Under this setting, we use TIHAD model to simulate diffusion as done in the previous experiments. We then compare cascade size distribution of items in I_1 against that of items in I_2. We also want to see if cascades generated by TIHAD are significantly different from those generated by a baseline diffusion model that does not consider item interaction. Hence, we generate another set of cascades following the same process using the LTLF model. The cascade size distributions of the two models are then compared.

Result: Figure 5.2 shows several interesting insights. First, it provides strong evidence that TIHAD model can capture the item interaction effect (among similar items) currently ignored by the existing models including LTLF. The figure shows that the cascade size of an item diffused with TIHAD is much larger than that of the item when it is diffused using LTLF. Moreover, the more similar an item with previous items, the larger cascade size it can reach. This makes sense since an item will receive more support in diffusion if it is more similar to other previously adopted items.

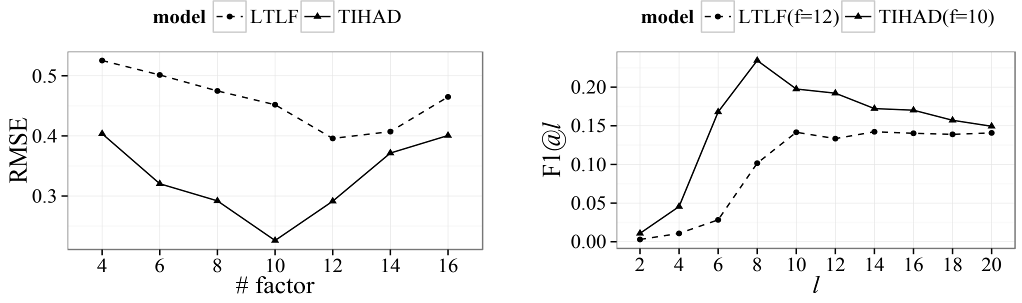
5.4.3 Hashtag Adoption Prediction Evaluation

This experiment aims to evaluate TIHAD using real dataset and compare it with the baseline LTLF model which does not consider item interaction.

Data set: We first collected the diffusion of hashtags in the Twitter network among Singapore users during on 14 February 2014, the Valentine Day. We expected that there should be some interesting diffusion cascades on this special day. We extracted the tweets of about 150,000 Singapore users from 3 to 16 February and sampled 1000 active users who adopted at least 3 hashtags per day. These users are connected by a social network with 9935 follow links.

We next wanted to determine the time step when each user first adopted a hashtag during the Valentine Day. Each time step duration is set as one hour. We confined ourselves to *fresh* hashtags which only appeared during Valentine Day but not the days during [3 Feb, 13 Feb]. We then identified the time step a user adopted a hashtag as the first time step in 14 February he used the hashtag. We obtained $20,847$ hashtags which the active users adopted from 00:00am to 11:59pm on the Valentine day. By filtering away unpopular hashtags, i.e., those with less than 5 active users adopting them, we were left with 4002 hashtags and $20,955$ adoptions. Based on Definition 13, we derived $100,250$ adoption labels (both adoption and non-adoption) associated with these 24 hours. Adoptions of the users on previous day (13 Feb) were used as their initial adoption histories. The hashtag diffusion data on 14 February from 0:00am to 11:59am is then used as the training data, while the remaining data on 14 February is used as the test data. The statistics of combined training and test datasets is summarized in Table 5.2.

Training process: We trained both TIHAD and LTLF using the diffusion training dataset on February 14. We tried different values for the regularization constant and observed that $\delta = 0.1$ gives the best result in terms of minimizing RMSE. We also tried different values for the number of recent items $k \in \{1, \ldots, 10\}$ and found that $k \in \{3, 4\}$ yield the best RMSE result for this training dataset. In the learning process, we observed that both models can achieve smallest RMSE for the training



(a) RMSE obtained on training data

(b) $F1@l$ obtained on test data

Figure 5.3: Comparing TIHAD against baseline LTLF. Both models were trained with regularization coefficient $\delta = 0.1$; for TIHAD, the number of recent items k is set as 3.

data.

Evaluation metrics: For evaluations, we used two accuracy metrics: (i) RMSE for measuring the model performance during training, and (ii) $F1@l$ when using the trained models for the *hashtag adoption prediction task* on the test data. To compute $F1@l$, we use the trained models to predict hashtag adoptions from 12:00 noon to 11:59pm of 14 Feb 2014. We selected those users who appear in both the training and test datasets and extracted from their tweets generated during the test period the hashtags that already appeared in the training set. The resultant test set had 884 users and 1219 hashtags.

Results: We first focus on the accuracy of trained models using RMSE defined on the training data. As shown in Figure 5.3a, the RMSE obtained by TIHAD is much smaller than that of LTLF when they are trained using the same dataset for different latent factor settings (i.e., $4 \leq f \leq 16$). TIHAD achieves the best RMSE when $f = 10$, while LTLF achieves best RMSE at $f = 12$.

In the hashtag adoption prediction task, TIHAD shows a huge improvement over LTLF as shown in Figure 5.3b. Other than $l = 2$, TIHAD outperforms LTLF for all other i values. The highest $F1$ achieved by TIHAD ($F1@8$) is more than 150% that of LTLF ($F1@10$).

As TIHAD achieves best accuracy at 10 factors, we would like to know what are the 10 factors. We manually check the top hashtags of each latent factor. We

Table 5.3: Latent factors and their top-3 hashtags

Latent Factors	Hashtags
Music bands/Singers	eminemftw, DatoSitiNurhaliza, SUL14
Local movies/actors	YouWhoCameFromTheStars, BrothersKeeper, GongLi
International movies/actors	frozen, jimmyfallon, KristenWiig
Music tour	RedAsiaTour, TheScriptUSTour, BANGERZTour2014
Sport	ICC2014, F1NightRace, LFCfacebook
Beauty	ILoveWTF, Dior, maybellinesg
Valentine	happyvalentine, firstvalentine, TweetforLove
Scandal/Controversy	AsylumSeekers, bigimmigrationrow, LittleIndiaRiot
Electronics	Xiaomi, ipadmini, Logitech
Self-improve	limitless, nickvijucic, empoweryourself

discover that the latent factors are topical and manually assign them topical labels. Table 5.3 shows the latent factors and their top 3 hashtags (due to limited space). Most of the latent factors (e.g., Music tour, Valentine, Electronics, Self-Improve) are self explanatory based on hashtags. The “Music bands/Singers” latent factor covers names of singers (e.g., Siti Nurhaliza and Eminem) and music concert (e.g., SUL14). The “Local movies/actors” latent factor covers popular movies (e.g., “You Who Came From the Stars”, “Brothers Keeper”) and actor (e.g., Gong Li). The other latent factors can be interpreted in a similar manner.

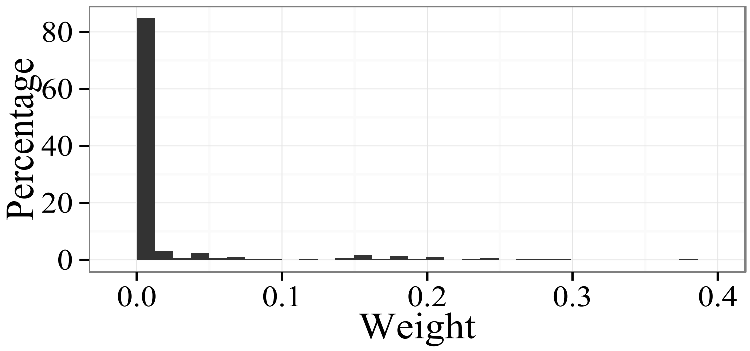


Figure 5.4: Histogram of influence weights $w_{v,u}$ which TIHAD learned for the network of Twitter users in our experiment.

Finally we would like to see what TIHAD can tell us about the network based on the homophily level and influence weights it learned. The homophily level learned by TIHAD is $h = 0.08$. This value is quite small and can be explained due to the sparseness of the network under study. Moreover the histogram of influence weights

$w_{v,u}$ in Figure 5.4 shows that most weights are very small (80% of them are close to 0), which matches the nature of weak links among most Twitter users.

5.5 Discussion

This work deals with the challenging problem of modeling diffusion with topic level interaction among items and homophily effect. The resultant model incorporates (i) item interaction by considering social influence from recent adoption events of user's neighbors, and (ii) homophily effect by modeling link weights as a linear function of user similarity. The behavior of the model under different settings and parameters have been investigated. The results on synthetic data show that both homophily and interaction at topic level can increase diffusion remarkably. Our experiments on hashtag diffusion on Twitter shows that TIHAD yields better prediction of hashtag diffusion on Twitter.

A promising direction for extension is to incorporate brand effects into the diffusion framework. The first possibility is to include also brand similarity between user and item. Specifically, we can adapt Equation 5.6 by adding a component of brand similarity, weighted by how much the user depends on brands in making her adoption decisions. The new equation is as follows.

$$\widehat{a}_{u,i,t} := \mu + b_u + b_i + \delta_u \theta_u^T \theta_i + (1-\delta_u) \overbrace{\beta_u^T \beta_i}^{brand\ similarity} + \sum_{v \in N_u} g(\theta_u^T \theta_v | h) \times \sum_j \theta_j^T \theta_i \tag{5.22}$$

In Equation 5.22, β_u and β_i represent respectively brand preference of user u and popularity of item i under different brands. The variable δ_u represents how much u depends on topic to make her adoption decisions. Another way to look at Equation 5.22, e.g. in the context of recommendation, is to consider two terms $\theta_u^T \theta_i$ and $\beta_u^T \beta_i$ as topic-based and brand-based ratings respectively.

The second possibility is to adapt item interaction to include brand-based interaction. Specifically, we may consider *intra*-brand interaction among items of the same brand and *inter*-brand interaction among items of different brands.

As the problem of jointly learning parameters is not convex, we are currently using grid search to deal with the non-convexity. However, the problem is still convex for each set of parameters if the others are kept fixed. Moreover, the current learning algorithm is still computationally expensive due to a large amount of operations required for computing gradients. As Alternative Least Square can handle both problems, we plan to use it to develop a better inference algorithm for our model.

Chapter 6

Conclusion

6.1 Dissertation Summary

In this section, we would like to summarize milestone accomplishments in this dissertation. Our research is motivated by the emergence of online social networks, social media and online shopping services which introduced new applications including viral marketing, item diffusion and recommendation systems. All these applications require *modeling of user-item adoption dynamics*, where "adoption dynamics" refers to the process of user making item adoption decisions. We started our study with the following research questions:

1. Among major factors with significant impact on item adoption decisions, which factors are overlooked in current literature? How to incorporate such factors into adoption dynamics modeling?

2. As socially connected users can influence one another to adopt items, how to incorporate such social effects into adoption dynamics modeling?

3. As item diffusion in real life usually involves several (dis)similar items — which may boost or impede each other's diffusion — how to model such interaction effects in item diffusion?

Chapter 3 provides answers for the first question. We find out that although topic-based user and item factors have been widely studied in existing literature, brand-based user and item factors are not. We thus propose two novel concepts, namely (i) *user brand-consciousness*, and (ii) *exclusiveness of item's brand*, and incorporate them into the so-called Brand-Item-Topic (BIT) model. We developed a Gibbs sampling inference for BIT which is able to infer successfully both topic-based and brand-based factors from adoption data without relying on prior information such as item price. Moreover, our empirical results show that brand exclusiveness and user brand-consciousness learned by BIT are indeed reasonable. We also evaluate BIT in item adoption prediction and show that BIT outperforms the baseline Latent Dirichlet Allocation, a well-known topic model.

Though BIT achieves promising results, we later found out that it is not scalable for large datasets due to a bottleneck in its inference process. We thus proposed a distributed and enhanced extension of BIT to remove this bottleneck and perform inference in a parallel manner. The resultant model, called DeBIT, is not only scalable to large datasets but also improves adoption prediction accuracy.

The success of BIT and DeBIT inspired us to adapt their ideas for answering the second question. In Chapter 4, we incorporated brand effects into existing social recommendation framework and proposed a novel model, called SocBIT. The inference of SocBIT was then formulated as a Maximum A Posteriori problem and solved by standard gradient descent method. We compared SocBIT with state-of-the-art social recommendation models such as SoRec [85] and RSTE [86] and showed that it outperforms the models significantly in item adoption prediction. Moreover, SocBIT's inference was shown to be efficient with a linear complexity of data size.

We provided an answer for the last question in Chapter 5 by proposing a general diffusion framework which captures topic-level interaction among items. We also integrated into this framework the homophily phenomenon — the tendency of users connecting to similar others — by modeling social influence between any two connected users as an *increasing* function of their similarity. Based on this framework, we developed the Topic level Interaction Homophily Aware Diffusion (TIHAD) model. TIHAD yields higher accuracy than baselines in an experiment on Twitter hashtag adoption prediction.

6.2 Future Work

To conclude this dissertation, we outline several promising research directions for further improvements of the current work.

- The diffusion framework in Chapter 5 can be extended further by incorporating brand effects. The first possibility is to include also brand similarity between user and item. The second possibility is to adapt item interaction to include brand-based interaction. Specifically, we may consider *intra*-brand interaction among items of the same brand and *inter*-brand interaction among items of different brands.

- In this dissertation, we modeled how supporting effect among similar items influences diffusion. A natural question is how to model impact of *item competition* on diffusion as well as adoption dynamics. Although some recent works [63, 102] have addressed the case of two competing items, the dynamics of diffusion when there are more than two competing items is still an open question. Moreover, item competition can also be combined with brand factors, e.g., by considering competition among items of different brands under the same topic.

- The *cold start* problem is not addressed in this dissertation. We thus want to explore how brand-related factors of cold start users or cold start items can be inferred. One possibility is to estimate brand-related factors of each such user (item) as the average of the corresponding factors of her (its) neighbors respectively. Requiring that brand-related factors of each user to be close to the corresponding averages of her neighbors also follows homophily. Finally, to satisfy this requirement, we can include a regularizer which hopefully can improve accuracy of the models and prevent overfitting.

Appendices

Appendix A

Modeling Brand Preference in Item Adoption

For easy following, we include here the notations of the models BIT and eBIT in Table A.1.

Table A.1: Notations for Brand-Item-Topic model

Notation	Description
U	Set of N users
I	Set of M items
B	Set of Q brands
Z	Set of K topics
$i_{u,n}$	Item at n-th adoption of user u
$z_{u,n}$	Latent topic of $i_{u,n}$
$b_{u,n}$	Latent brand of $i_{u,n}$
$d_{u,n}$	Latent decision variable of this adoption
ω_b	Parameters for the item distribution of brand b
β	Hyperparameter for Dirichlet prior of ω_b
ψ_z	Parameters for the brand distribution of topic z
α	Hyperparameters for Dirichlet prior of ψ_z
ϑ_u	Parameters for the topic distribution of user u
θ	Hyperparameters for Dirichlet prior of ϑ_u
φ_z	Parameters for the item distribution of topic z
ϕ	Hyper parameters for Dirichlet prior of φ_z's
δ_u	Parameters for binomial distribution of $d_{u,n}$
γ	Hyper parameter for sampling Λ
h	Vector of hyperparameters, e.g. $h = (\alpha, \beta, \theta, \phi, \gamma)$

In the following parts, 0-decision (1-decision) corresponds to topic-based (brand-based) adoptions resp. Moreover, topic-based adoptions with no brand involved can be considered as a special brand-based adoption where the brand is *empty* brand. From now on, we use value 0 to denote empty brand ($b = 0$ means there is no brand).

A.1 Propositions

A.1.1 Sampling strategy

Challenge: Naive sampling latent variables in a sequential manner is actually *wrong*. Indeed, for a given adoption $j = (u, n)$, sampling decision d_j and brand b_j variables separately can lead to the case $d_j = 0$ while $b_j \neq 0$. This is a contradiction as the corresponding adoption is both topic-based and brand-based at the same time!

Solution: Thus, for each j-th adoption, we will *sample simultaneously* d_j and b_j as a latent pair $y_j = (d_j,\ b_j)$. When $d_j = 0$, we have a topic-based adoption, thus b_j must be 0 and $y_j = (0,0) = 0$ (the only pair for topic-based adoptions). When $d_j = 1$, b_j must be some brand b and $y_j = (1, b)$. In short, we define a new kind of latent variable y which receives values in the following set.

$$S = \{(0,0), (1,b_1),\ \ldots,\ (1,b_Q)\}$$

Given this new kind of variable, the generative process can be re-drawn as in Fig. A.1. Please note that this alternative is still equivalent to the original representation of BIT. Here we just use it to ease explanation of Gibbs sampling with new latent variables y.

Figure A.1 reveals two dependencies:

1. Nodes y now depend on nodes u and z. More precisely, decision component depends on decision distribution of u and brand component depends on brand distribution of z.

2. Item nodes now depend on topic nodes z and nodes y.

Thus, we need to combine original user-decision and topic-brand distributions into a coupled distribution for drawing y's. We also need to clarify how to draw an item node i given its parent nodes z and y. All these are given in the following definition.

Definition 16. *Distributions to draw y nodes and item nodes in alternative BIT can be linked to distributions in original BIT as follows.*

D1) *The conditional probability of drawing a node $y = (d, b)$ given user and topic is given by*

$$p(y = (d,\ b)|u,z) = \begin{cases} p(d = 0|u) = \delta_{u,0}, & \text{if } (d,\ b) = (0,0) \\ p(d = 1|u) \times p(b|z) = \delta_{u,1} \times \psi_{z,b}, & \text{if } (d,\ b) = (1,b) \end{cases} \tag{A.1}$$

D2) *The conditional probability of drawing an item node i given its parents z and y is given by*

$$p(i|z,y) = \begin{cases} p(i|z) = \varphi_{z,i}, & \text{if } y = (0,0), \textit{ (topic-based)} \\ p(i|b) = \omega_{b,i}, & \text{if } y = (1,b), \textit{ (brand-based)} \end{cases} \tag{A.2}$$

For easy following, all notations necessary for Gibbs sampling are summarized in Table A.2. With all these preparations, we are ready to state our propositions.

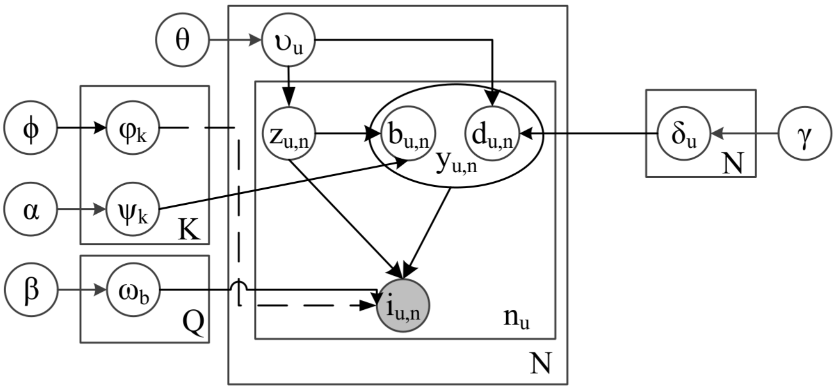


Figure A.1: Alternative representation for generative process of BIT where decision and brand variables are *coupled*

Table A.2: Notations used in training BIT

Notation	Description
n_u^0 and n_u^1	Number of *topic*-based and *brand*-based adoptions of u.
$n_u = n_u^0 + n_u^1$	Total number of adoptions of user u.
$\mathbf{c}_u = (c_{u,z_1}, \ldots, c_{u,z_K})$	Number of times topics assigned to u.
$dc_u = (dc_{u,0}, dc_{u,1})$	Number of times decisions assigned to u.
$c_z^u = (c^u_{z,\, b_1}, \ \ldots, c^u_{z,\, b_Q})$	Number of times brands assigned to u under topic z.
$i_u = \{i_{u,1}, \ \ldots, i_{u,n_u}\}$	Adoptions by user u.
$z_u = \{z_{u,1}, \ \ldots, z_{u,n_u}\}$	Latent topics of these adoptions.
$y_u = \{y_{u,1}, \ \ldots, \ y_{u,n_u}\}$	Latent pairs $y_j = (d_j, \ b_j)$ of these adoptions.
$\mathscr{I} = \bigcup_u i_u$	Set of *observed* adoptions from all users (fixed).
$\mathscr{Z} = \bigcup_u z_u$	Set of all latent topics (changes each iteration).
$\mathscr{Y} = \bigcup_u y_u$	Set of all latent pairs (changes each iteration).
$c_z = (c_{z,i_1}, \ \ldots, c_{z,i_M})$	Number of times items assigned to z *by topic-based* adoptions.
$bc_z = \left(c_{z,b_1}, \ \ldots, c_{z,b_Q}\right)$	Number of times brands assigned to z by *brand-based* adoptions.
$c_b = (c_{b,i_1}, \ \ldots, c_{b,i_M})$	Number of times items assigned to brand b by *brand-based* adoptions.
$\mathscr{U}$	Vector of users for all adoptions.

A.1.2 Proposition statements

Recall that observed data is denoted as $\mathscr{I}$ and $\mathscr{U}$ as in Table A.2.

For each adoption $j = (\tilde{u}, n)$ where adopted item is $\tilde{i}$, we state

- how to sample *latent topic* z_j given remaining latent variables $\mathscr{Z}^{-j}$, $\mathscr{Y}$.
- how to sample *latent coupled variable* y_j given remaining latent variables $\mathscr{Y}^{-j}$, $\mathscr{Z}$

here $\mathscr{Z}^{-j} = \mathscr{Z} - \{z_j\}$ and $\mathscr{Y}^{-j} = \mathscr{Y} - \{y_j\}$.

Proposition 5 (Updating latent topic)**.** *Given remaining latent variables $\mathscr{Z}^{-j}$, $\mathscr{Y}$ and data, we can sample a value for latent topic z_j based on whether the j-th adoption is topic-based or brand-based.*

C1. (Topic-based adoption) If currently $y_j = 0$, the sampling distribution is proportional to the following $w_0^*(.)$ function.

$$P(z_j = z | \mathscr{Z}^{-j}, \ \mathscr{Y}) \propto w_0^*(z) := \frac{c_{\tilde{u},\, z} + \theta - 1}{n_{\tilde{u}} + K\theta - 1} \times \frac{dc_{\tilde{u},0} + \gamma - 1}{n_{\tilde{u}} + 2\gamma - 1} \times \frac{c_{z,\, \tilde{i}} + \phi - 1}{\sum_i c_{z,\, i} + M\phi - 1} \tag{A.3}$$

Since terms $dc_{\tilde{u},0} + \gamma - 1$, $n_{\tilde{u}} + K\theta - 1$ and $n_{\tilde{u}} + 2\gamma - 1$ do not depend on topic, we can remove them and finally get a simpler version for weights

$$P(z_j = z | \mathscr{Z}^{-j}, \mathscr{Y}) \propto w_0(z) := (c_{\tilde{u},\, z} + \theta - 1) \times \frac{c_{z,\, \tilde{i}} + \phi - 1}{\sum_i c_{z,\, i} + M\phi - 1} \tag{A.4}$$

This simpler version provides an intuitive explanation that for a topic-based adoption, the most likely topic z should "match" both user and item of the adoption.

C2. *(Brand-based adoption) If currently $y_j = (1, \tilde{b})$, sampling distribution is proportional to the following $w_1^*(.)$ function.*

$$\begin{aligned} P(z_j = z | \mathscr{Z}^{-j}, \mathscr{Y}) \propto w_1^*(z) := & \frac{c_{\tilde{u},\, z} + \theta - 1}{n_{\tilde{u}} + K\theta - 1} \times \frac{dc_{\tilde{u},1} + \gamma - 1}{n_{\tilde{u}} + 2\gamma - 1} \times \\ & \frac{c_{z,\, \tilde{b}} + \alpha - 1}{\sum_b c_{z,b} + Q\alpha - 1} \times \frac{c_{\tilde{b},\, \tilde{i}} + \beta - 1}{\sum_i c_{\tilde{b},\, i} + M\beta - 1} \end{aligned} \tag{A.5}$$

By removing terms independent from topic, we finally get

$$P(z_j = z | \mathscr{Z}^{-j}, \mathscr{Y}) \propto w_1(z) := (c_{\tilde{u},\, z} + \theta - 1) \times \frac{c_{z,\, \tilde{b}} + \alpha - 1}{\sum_b c_{z,b} + Q\alpha - 1} \tag{A.6}$$

This simpler version provides an intuitive explanation that for a brand-based adoption, the most likely topic z should "match" both user and brand of the adoption.

Now to the proposition for updating latent pair $y_j = (d_j,\, b_j)$ for j-th adoption. Note that now the topic z_j has a *known* value $\tilde{z}$.

Proposition 6 (Updating latent variables y)**.** *Given remaining latent variables $\mathscr{Y}^{-j}$, $\mathscr{Z}$ and data, we sample a value $\tilde{y}$ for latent y_j using the distribution which is propor-*

tional to the following $w_y^*(.)$ function.

$$P(y_j = y|\mathscr{Y}^{-j},\ \mathscr{X}) \propto$$
$$w_y^*(y) := \begin{cases} \dfrac{dc_{\tilde{u},0}+\gamma-1}{n_{\tilde{u}}+2\gamma-1} \times \dfrac{c_{\tilde{z},\ \tilde{i}}+\phi-1}{\sum_i c_{\tilde{z},\ i}+M\phi-1}, & \text{if } y=0 \\ \\ \dfrac{dc_{\tilde{u},1}+\gamma-1}{n_{\tilde{u}}+2\gamma-1} \times \dfrac{c_{\tilde{z},\ b}+\alpha-1}{\sum_b c_{\tilde{z},b}+Q\alpha-1} \times \dfrac{c_{b,\ \tilde{i}}+\beta-1}{\sum_i c_{b,i}+M\beta-1}, & \text{if } y=(1,\ b) \end{cases} \tag{A.7}$$

We can remove the term $n_{\tilde{u}}+2\gamma-1$, which is common for both cases, and get a simpler version for weights as follows.

$$P(y_j = y|\mathscr{Y}^{-j},\ \mathscr{X}) \propto$$
$$w_y(y) := \begin{cases} (dc_{\tilde{u},0}+\gamma-1) \times \dfrac{c_{\tilde{z},\ \tilde{i}}+\phi-1}{\sum_i c_{\tilde{z},\ i}+M\phi-1}, & \text{if } y=0 \\ \\ (dc_{\tilde{u},1}+\gamma-1) \times \dfrac{c_{\tilde{z},\ b}+\alpha-1}{\sum_b c_{\tilde{z},b}+Q\alpha-1} \times \dfrac{c_{b,\ \tilde{i}}+\beta-1}{\sum_i c_{b,i}+M\beta-1}, & \text{if } y=(1,b) \end{cases} \tag{A.8}$$

A.2 Proofs of propositions

The following proofs follow closely the spirit of the work [48]. We recall some useful formula from the work.

A.2.1 Dirichlet distribution

For a given parameter vector $\alpha = (\alpha_1,\ \ldots,\ \alpha_d) \in R^d$, the Dirichlet distribution over d-dimension probability simplex has the following formula.

For a given point $p = (p_1,\ \ldots,\ p_d)$ in d-dimension probability simplex, the probability of drawing the point p is given by

$$p(p|\alpha) = Dir(p|\alpha) := \frac{1}{\Delta(\alpha)} \prod_i p_i^{\alpha_i - 1} \tag{A.9}$$

where $\Delta(\alpha)$ is the normalization constant and determined by

$$\Delta(\alpha) = \frac{\prod_i \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)} \tag{A.10}$$

where $\Gamma()$ denotes the standard Gamma function.

In this work, all *priors* are symmetric Dirichlet distribution where all components of α is the same scalar α i.e. $\alpha = (\alpha,\ \ldots,\ \alpha)$.

To derive the conditional probabilities in our propositions, we first need to look at the joint probability.

A.2.2 Joint probability

Due to dependencies among variables in our model, the joint probability $P\left(\mathscr{I},\ \mathscr{Y},\ \mathscr{Z},\ \mathscr{U}\right)$ can be rewritten as

$$P\left(\mathscr{Y},\ \mathscr{Z},\ \mathscr{I},\ \mathscr{U}\right) = P(\mathscr{I}|\mathscr{Y},\ \mathscr{Z}) \times P(\mathscr{Y}|\mathscr{U}, \mathscr{Z}) \times P(\mathscr{Z}|\mathscr{U}) \times P(\mathscr{U}) \tag{A.11}$$

Using this equation of joint probability, we can derive the two following lemmas, which help in our proofs.

Lemma 1. *We claim that*

$$P(y_j = \tilde{y}|\mathscr{Y}^{-j},\ \mathscr{Z},\ \mathscr{I},\ \mathscr{U}) \propto \frac{P(\mathscr{I}|\mathscr{Y}(\tilde{y}),\ \mathscr{Z})}{P(\mathscr{I}^{-j}|\mathscr{Y}^{-j},\ \mathscr{Z}^{-j})} \times \frac{P(\mathscr{Y}(\tilde{y})|\mathscr{U}, \mathscr{Z})}{P(\mathscr{Y}^{-j}|\mathscr{U}^{-j}, \mathscr{Z}^{-j})} \tag{A.12}$$

where $\mathscr{Y}(\tilde{y}) = \mathscr{Y}^{-j} \cup \{\tilde{y}\}$ *changes with different values of* $\tilde{y}$

Lemma 2. *We claim that*

$$P(z_j = \tilde{z} | \mathscr{Z}^{-j}, \mathscr{Y}, \mathscr{I}, \mathscr{U}) \propto \frac{P[\mathscr{I} | \mathscr{Y}, \mathscr{Z}(\tilde{z})]}{P(\mathscr{I}^{-j} | \mathscr{Y}^{-j},\ \mathscr{Z}^{-j})} \times \frac{P[\mathscr{Y} | \mathscr{U}, \mathscr{Z}(\tilde{z})]}{P(\mathscr{Y}^{-j} | \mathscr{U}^{-j},\ \mathscr{Z}^{-j})} \times \frac{P[\mathscr{Z}(\tilde{z}) | \mathscr{U}]}{P(\mathscr{Z}^{-j} | \mathscr{U}^{-j})} \tag{A.13}$$

where $\mathscr{Z}(\tilde{z}) = \mathscr{Z}^{-j} \cup \{\tilde{z}\}$ changes with different values of $\tilde{z}$.

Proof of Lemma 1

Proof. Indeed, we have

$$\begin{aligned}
P(y_j = \tilde{y} | \mathscr{Y}^{-j},\ \mathscr{Z},\ \mathscr{I},\ \mathscr{U}) &= \frac{P(\mathscr{Y}(\tilde{y}),\ \mathscr{Z},\ \mathscr{I},\ \mathscr{U})}{P(\mathscr{Y}^{-j},\ \mathscr{Z},\ \mathscr{I},\ \mathscr{U})} \\
&= \frac{P(\mathscr{Y}(\tilde{y}),\ \mathscr{Z},\ \mathscr{I},\ \mathscr{U})}{P(\mathscr{Y}^{-j},\ \mathscr{Z}^{-j},\ \mathscr{I}^{-j},\ \mathscr{U}^{-j}) \times P(z_j, i_j, u_j)} \\
&= \frac{P(\mathscr{Y}(\tilde{y}),\ \mathscr{Z},\ \mathscr{I},\ \mathscr{U})}{P(\mathscr{Y}^{-j},\ \mathscr{Z}^{-j},\ \mathscr{I}^{-j},\ \mathscr{U}^{-j}) \times P(\tilde{z}, \tilde{i}, \tilde{u})} \\
&\propto \frac{P(\mathscr{Y}(\tilde{y}),\ \mathscr{Z},\ \mathscr{I},\ \mathscr{U})}{P(\mathscr{Y}^{-j},\ \mathscr{Z}^{-j},\ \mathscr{I}^{-j},\ \mathscr{U}^{-j})}
\end{aligned} \tag{A.14}$$

here $\mathscr{Y}(\tilde{y}) = \mathscr{Y}^{-j} \cup \tilde{y}$. The last equation is obtained by noting that $\tilde{z}, \tilde{i}, \tilde{u}$ are fixed values (variable y_j is the only thing changes here).

Applying Eqn. (A.11) for both nominator and denominator on RHS of Eqn. (A.14) and noting that the *last two term can be skipped* (as they are independent from $\mathscr{Y}$), we obtain Eqn. (A.12). □

Proof of Lemma 2

Proof. For this lemma, it should be noted that $\mathscr{Z}$ is now the changing component instead of $\mathscr{Y}$. In fact, $\mathscr{Z}$ changes with $\tilde{z}$ since $\mathscr{Z} = \mathscr{Z}(\tilde{z}) = \mathscr{Z}^{-j} \cup \tilde{z}$. By proceeding similarly as Eqn. (A.14), we can obtain the following formula.

$$P(z_j = \tilde{z} | \mathscr{Z}^{-j}, \mathscr{Y}, \mathscr{I}, \mathscr{U}) \propto \frac{P(\mathscr{Z}(\tilde{z}), \mathscr{Y}, \mathscr{I}, \mathscr{U})}{P(\ \mathscr{Z}^{-j},\ \mathscr{Y}^{-j},\ \mathscr{I}^{-j},\ \mathscr{U}^{-j})} \tag{A.15}$$

Applying again Eqn. (A.11) for joint probabilities in nominator and denominator, (for nominator, noting that $P(\mathscr{U})$ can be skipped since it is independent from

$\mathscr{Z}(\tilde{z})$), we have

$$P(\mathscr{Z}(\tilde{z}),\mathscr{Y},\mathscr{I},\mathscr{U}) \propto P[\mathscr{I}|\mathscr{Y},\mathscr{Z}(\tilde{z})] \times P[\mathscr{Y}|\mathscr{U},\mathscr{Z}(\tilde{z})] \times P[\mathscr{Z}(\tilde{z})|\mathscr{U}] \quad \text{(A.16a)}$$

$$P\left(\mathscr{Z}^{-j},\ \mathscr{Y}^{-j},\ \mathscr{I}^{-j},\ \mathscr{U}^{-j}\right) \propto P(\mathscr{I}^{-j}|\mathscr{Y}^{-j},\ \mathscr{Z}^{-j}) \times P(\mathscr{Y}^{-j}|\mathscr{U}^{-j},\ \mathscr{Z}^{-j}) \times P(\mathscr{Z}^{-j}|\mathscr{U}^{-j}) \quad \text{(A.16b)}$$

Combining Eqns. (A.15), (A.16a) and (A.16b), we obtain Eqn. (A.13). □

Comparing Eqns. (A.12) and (A.13), we find that quantities $P(\mathscr{I}|\mathscr{Y},\ \mathscr{Z})$, $P(\mathscr{Y}|\mathscr{U},\ \mathscr{Z})$ and $P(\mathscr{Z}|\mathscr{U})$ are common. Thus we need to derive equations for them. These derivations are provided below.

A.2.3 Common expressions

We collect all necessary lemmas below.

Lemma 3.

$$P(\mathscr{Z}|\mathscr{U}) = \prod_{u \in U} \frac{\Delta(c_u + \theta)}{\Delta(\theta)} \quad \text{(A.17)}$$

Lemma 4.

$$P(\mathscr{I}|\mathscr{Y},\ \mathscr{Z}) = \left[\prod_{z} \frac{\Delta(c_z + \phi)}{\Delta_M(\phi)}\right] \times \left[\prod_{b} \frac{\Delta(c_b + \beta)}{\Delta_M(\beta)}\right] \quad \text{(A.18)}$$

Lemma 5.

$$P(\mathscr{Y}|\mathscr{U},\ \mathscr{Z}) = \prod_{u} \left\{ \frac{\Delta(dc_u + \gamma)}{\Delta_2(\gamma)} \times \prod_{z} \frac{\Delta\left(c^u_z + \alpha\right)}{\Delta_Q(\alpha)} \right\} \quad \text{(A.19)}$$

where $c^u_z = (c^u_{z,b_1},\ \ldots,\ c^u_{z,b_Q})$ contains frequencies at which brands are chosen after user u chose topic z.

Proving Lemma 3

Proof. This term is exactly the same as its counterpart in LDA. Thus, we just reuse the formula (72) in [48]. □

Proving Lemma 4

Proof. Due to independence among topic-based and brand-based adoptions, we can split $P(\mathscr{I}|\mathscr{Z},\ \mathscr{Y})$ as follows.

$$P(\mathscr{I}|\mathscr{Z},\ \mathscr{Y}) = P(\mathscr{I}^0|\mathscr{Z}^0) \times P(\mathscr{I}^1|\mathscr{B}) \tag{A.20}$$

where

- $\mathscr{I}^0$ and $\mathscr{I}^1$ contain items from topic-based and brand-based adoptions resp
- $\mathscr{Z}^0$ contains topics wrt $\mathscr{I}^0$.
- $\mathscr{B}$ contains brands in brand-based adoptons.

Now the term $P(\mathscr{I}^0|\mathscr{Z}^0)$ is exactly the same as its LDA counterpart, the probability of word vector given topic vector $P(w|z)$. Thus, we just adapt formula (68) in [48] using our notations and obtain

$$P(\mathscr{I}^0|\mathscr{Z}^0) = \left[\prod_z \frac{\Delta(c_z + \phi)}{\Delta_M(\phi)}\right] \tag{A.21}$$

Similarly, formula for the term $P(\mathscr{I}^1|\mathscr{B})$ corresponding to brand-based adoptions can be easily obtained by using brands in place of topics. Thus, we get

$$P(\mathscr{I}^1|\mathscr{B}) = \left[\prod_b \frac{\Delta(c_b + \beta)}{\Delta_M(\beta)}\right] \tag{A.22}$$

Combining Eqns. (A.20), (A.21) and (A.22) finish the proof. □

Proving Lemma 5

Proof. We have

$$\begin{aligned} P(\mathscr{Y}|\mathscr{U},\ \mathscr{Z}) &= \prod_u P(y_u|z_u) \\ &= \prod_u \left[\int_{\delta_u} P(y_u|z_u, \delta_u) P(\delta_u) d\delta_u\right] \end{aligned} \tag{A.23}$$

To estimate the integral for each user, we need to divide his latent variables y_u into two parts y_u^0 and y_u^1. The former (the latter) corresponds to topic-based (brand-based) adoptions resp. Due to conditional independence between y_u^0 and y_u^1 given parameters δ_u, we have

$$\begin{aligned}\int_{\delta_u} P(y_u|z_u,\delta_u)P(\delta_u) &= \int_{\delta_u} P(y_u^0|\delta_u) \times P(y_u^1|\delta_u,z_u^1) \times P(\delta_u)d\delta_u \\ &= \int_{\delta_u} \delta_{u,0}^{c_{u,0}} \times P(y_u^1|\delta_u,z_u^1) \times P(\delta_u)d\delta_u \qquad \text{(A.24)}\end{aligned}$$

where we replaced $P(y_u^0|\delta_u)$ by $\delta_{u,0}^{c_{u,0}}$ (easy).

The hard part is the factor $P(y_u^1|\delta_u,z_u^1)$. Since a node y also depends on topic, we need to split vector y_u^1 into sub-vectors, each of the same topic. Precisely, we use the following partition

$$y_u^1 = \cup_{z \in Z} y_{u,z}$$

where $y_{u,z} = \{(1,b_{i_1}),\ \ldots,\ (1,b_{i_l})\}$ is a vector of latent pairs under the same topic node z.

Due to this partition, we get

$$\begin{aligned}P(y_u^1|\delta_u,z_u^1) &= \prod_z \left[P(y_{u,z}^1|\delta_u,\ z)\right] \\ &= \prod_z \left[\int_{\psi_z} P(y_{u,z}^1|\delta_u,\ \psi_z) \times P(\psi_z)d\psi_z\right] \\ &= \prod_z \left[\delta_{u,1}^{l_z} \frac{\int_{\psi_z} \prod_b \psi_{z,b}^{c_{z,b}^u+\alpha-1} d\psi_z}{\Delta_Q(\alpha)}\right] \qquad (l_z:\ \text{length of } y_{u,z}^1) \\ &= \prod_z \left[\delta_{u,1}^{l_z} \frac{\Delta(c_z^u+\alpha)}{\Delta_Q(\alpha)}\right] \qquad \text{(A.25)}\end{aligned}$$

where in the last two equations we used the formula of Dirichlet prior $P(\psi_z)$ and Dirichlet integral resp.

We finish the proof of this lemma by the following steps:

- Substituting Dirichlet prior $P(\delta_u)$ and Eqn. (A.25) into Eqn. (A.24)
- Collapsing $\prod_z \delta_{u,1}^{l_z}$ into $\delta_{u,1}^{c_{u,1}}$ (since $\sum_z l_z = c_{u,1}$)
- Using Dirichlet integral over decision δ_u

Indeed, we have

$$\begin{aligned}\int_{\delta_u} P(y_u|z_u,\delta_u)P(\delta_u) &= \left\{ \int_{\delta_u} \frac{\delta_{u,0}^{c_{u,0}+\gamma-1} \cdot \delta_{u,1}^{c_{u,1}+\gamma-1}}{\Delta_2(\gamma)} \right\} \times \prod_z \frac{\Delta(c_z^u+\alpha)}{\Delta_Q(\alpha)} \\ &= \frac{\Delta(dc_u+\gamma)}{\Delta_2(\gamma)} \times \prod_z \frac{\Delta(c_z^u+\alpha)}{\Delta_Q(\alpha)}\end{aligned}$$

□

Now that all necessary formulae are ready, we proceed to proving our two propositions.

A.2.4 Proof for Prop. 6

Now we need formulae for first and second fractions in Eqn. (A.12). They are summarized in the following lemma.

Lemma 6. *Given that $z_j = \tilde{z}$, $i_j = \tilde{i}$ and $u_j = \tilde{u}$, we have*

1. *(First fraction)*

$$\frac{P(\mathscr{I}|\mathscr{Y}(\tilde{y}),\ \mathscr{Z})}{P(\mathscr{I}^{-j}|\mathscr{Y}^{-j},\ \mathscr{Z}^{-j})} = \begin{cases} \dfrac{c_{\tilde{z},\ \tilde{i}}+\phi-1}{\sum_i c_{\tilde{z},\ i}+M\phi-1}, & \text{if } \tilde{y}=(0,0) \\ \dfrac{c_{\tilde{b},\ \tilde{i}}+\beta-1}{\sum_i c_{\tilde{b},\ i}+M\beta-1}, & \text{if } \tilde{y}=(1,\tilde{b}) \end{cases} \tag{A.26}$$

2. *(Second fraction)*

$$\frac{P(\mathscr{Y}(\tilde{y})|\mathscr{U},\mathscr{Z})}{P(\mathscr{Y}^{-j}|\mathscr{U}^{-j},\mathscr{Z}^{-j})} = \begin{cases} \dfrac{dc_{\tilde{u},\ 0}+\gamma-1}{n_{\tilde{u}}+2\gamma-1}, & \text{if } \tilde{y}=(0,0) \\ \dfrac{dc_{\tilde{u},\ 1}+\gamma-1}{n_{\tilde{u}}+2\gamma-1} \times \dfrac{c_{\tilde{z},\ \tilde{b}}^{\tilde{u}}+\alpha-1}{c_{\tilde{u},\ \tilde{z}}^{1}+Q\alpha-1}, & \text{if } \tilde{y}=(1,\tilde{b}) \end{cases} \tag{A.27}$$

Obviously, once the two formulae is established, Prop. 6 is a straightforward combination of them. Thus, we actually only need to prove the two formulae as follows.

Proof. **(First fraction)**

Applying Eqn. (A.18) for $(\mathscr{I},\ \mathscr{Y}(\tilde{y}),\ \mathscr{Z})$ and $(\mathscr{I}^{-j},\ \mathscr{Y}^{-j},\ \mathscr{Z}^{-j})$ respectively, we get

$$\frac{P(\mathscr{I}|\mathscr{Y}(\tilde{y}),\ \mathscr{Z})}{P(\mathscr{I}^{-j}|\mathscr{Y}^{-j},\ \mathscr{Z}^{-j})} = \left[\prod_z \frac{\Delta(c_z(\tilde{y})+\phi)}{\Delta(c_z^{-j}+\phi)}\right] \times \left[\prod_b \frac{\Delta(c_b(\tilde{y})+\beta)}{\Delta(c_b^{-j}+\beta)}\right] \tag{A.28}$$

In these fractions, there is actually only one difference coming from removing j-th adoption. Thus, we need to examine this adoption. Keep in mind that we already know values of z_j, i_j and u_j, which we denote by $\tilde{z}, \tilde{i}$ and $\tilde{u}$ resp. Depending on the value of $\tilde{y}$, we have the following cases.

C1) If $\tilde{y} = (0,0)$ then the adoption is topic-based. Thus, removing it has no effect on brand-based part. Hence the second product in Eqn. (A.28) will be cancelled out and only first product remains. Moreover, in the first product, only the counts of topic $\tilde{z}$ is affected. Thus, factors of other topic will be cancelled and only the factor of $\tilde{z}$ remains. In short, the whole RHS of (A.28) collapsed to only one following factor.

$$\frac{\Delta(c_{\tilde{z}}+\phi)}{\Delta(c_{\tilde{z}}^{-j}+\phi)}$$

For this factor, we have

- $c_{\tilde{z},\ \tilde{i}}^{-j} = c_{\tilde{z},\ \tilde{i}} - 1$

  AND

- $c_{\tilde{z},i}^{-j} = c_{\tilde{z},i}$ for $i \neq \tilde{i}$

Thus, on applying Eqn. (A.10) of $\Delta()$ function, we get

- $\dfrac{\Gamma(c_{\tilde{z},\ i})}{\Gamma(c_{\tilde{z},\ i}^{-j})} = 1, \forall i \neq \tilde{i}$

- $\dfrac{\Gamma(c_{\tilde{z},\,\tilde{i}})}{\Gamma(c_{\tilde{z},\,\tilde{i}}^{-j})} = c_{\tilde{z},\,\tilde{i}} + \phi - 1$
- $\dfrac{\Gamma(M\phi + \sum_i c_{\tilde{z},\,i} - 1)}{\Gamma(M\phi + \sum_i c_{\tilde{z},\,i})} = \dfrac{1}{M\phi + \sum_i c_{\tilde{z},\,i} - 1}$

Hence, for the first case, we finally get

$$\begin{aligned}\frac{P(\mathscr{I}|\mathscr{Y}(\tilde{y}),\ \mathscr{Z})}{P(\mathscr{I}^{-j}|\mathscr{Y}^{-j},\ \mathscr{Z}^{-j})} &= \frac{\Delta(c_{\tilde{z}} + \phi)}{\Delta(c_{\tilde{z}}^{-j} + \phi)}\\ &= \frac{\Gamma(c_{\tilde{z},\,\tilde{i}})}{\Gamma(c_{\tilde{z},\,\tilde{i}}^{-j})} \times \frac{\Gamma(M\phi + \sum_i c_{\tilde{z},\,i} - 1)}{\Gamma(M\phi + \sum_i c_{\tilde{z},\,i})}\\ &= \frac{c_{\tilde{z},\,\tilde{i}} + \phi - 1}{\sum_i c_{\tilde{z},\,i} + M\phi - 1}\end{aligned}$$

This is exactly what we want.

C2) If $\tilde{y} = (1, \tilde{b})$ then the adoption is *brand*-based. Thus, now the first product (topic-based part) will be cancelled out and second product remains. For second product, we can use the same argument as the first case, with brand in place of topic. Hence, we also obtain what we want.

□

Now we move to proof of second fraction.

Proof. **(Second fraction)**

Applying Eqn. (A.19) in Lemma 5 for $(\mathscr{Y}(\tilde{y}),\ \mathscr{U},\ \mathscr{Z})$ and $(\mathscr{Y}^{-j},\ \mathscr{U}^{-j},\ \mathscr{Z}^{-j})$ resp. and note that removing j-th adoption only affects quantities of user $\tilde{u}$, we get

$$\begin{aligned}\frac{P(\mathscr{Y}(\tilde{y})|\mathscr{U},\mathscr{Z})}{P(\mathscr{Y}^{-j}|\mathscr{U}^{-j},\mathscr{Z}^{-j})} &= \frac{\Delta(dc_{\tilde{u}} + \gamma)}{\Delta(dc_{\tilde{u}}^{-j} + \gamma)} \times \prod_z \frac{\Delta(c_z^{\tilde{u}} + \alpha)}{\Delta(c_z^{\tilde{u},-j} + \alpha)}\\ &= \frac{\Delta(dc_{\tilde{u}} + \gamma)}{\Delta(dc_{\tilde{u}}^{-j} + \gamma)} \times \frac{\Delta(c_{\tilde{z}}^{\tilde{u}} + \alpha)}{\Delta(c_{\tilde{z}}^{\tilde{u},-j} + \alpha)} \qquad \text{(A.29)}\end{aligned}$$

where in the last equality we used the fact that $c_z^{\tilde{u}} = c_z^{\tilde{u},-j}$ for $z \neq \tilde{z}$ to cancel all factors of topics other than $\tilde{z}$.

Again, depend on value of $\tilde{y}$, we have following cases.

C1) If $\tilde{y} = (0,0)$ then adoption is topic-based. Thus, *second factor is cancelled out* (since it has nothing to do with topic-based adoption). Moreover, we also have $dc_{\tilde{u},0}^{-j} = dc_{\tilde{u},0} - 1$ and $dc_{\tilde{u},1}^{-j} = dc_{\tilde{u},1}$. Thus, by formula of Δ function, the first factor can be reduced to the fraction

$$\frac{dc_{\tilde{u},0} + \gamma - 1}{n_{\tilde{u}} + 2\gamma - 1}$$

In short, the whole RHS of (A.29) collapse to this fraction, which is exactly desired result.

C2) If $\tilde{y} = (1, \tilde{b})$ then adoption is brand-based. Thus, we now have $dc_{\tilde{u},1}^{-j} = dc_{\tilde{u},1} - 1$ and $dc_{\tilde{u},0}^{-j} = dc_{\tilde{u},0}$; which reduces the first factor to

$$\frac{dc_{\tilde{u},1} + \gamma - 1}{n_{\tilde{u}} + 2\gamma - 1}$$

The second factor can be easily derived as

$$\frac{c_{\tilde{z},\, \tilde{b}}^{\tilde{u}} + \alpha - 1}{c_{\tilde{u},\, \tilde{z}}^{1} + Q\alpha - 1}$$

Multiplying the two fractions gives us desired result.

□

A.2.5 Proof for Prop. 5

Now we need to compute the three fractions on the RHS of Eqn. (A.13). The results are summarized in the following lemma.

Lemma 7. *Given $i_j = \tilde{i}$, $u_j = \tilde{u}$ and value $\tilde{y}$ of pair variable y_j, we can compute the three fractions as follows.*

1. *(Ratio of cond. probs. for topic)*

$$\frac{P[\mathscr{Z}(\tilde{z})|\mathscr{U}]}{P(\mathscr{Z}^{-j}|\mathscr{U}^{-j})} = \frac{c_{\tilde{u},\, \tilde{z}} + \theta - 1}{n_{\tilde{u}} + K\theta - 1} \tag{A.30}$$

2. *(Ratio of cond. probs. for item)*

$$\frac{P[\mathscr{I}|\mathscr{Y},\mathscr{Z}(\tilde{z})]}{P(\mathscr{I}^{-j}|\mathscr{Y}^{-j},\ \mathscr{Z}^{-j})} = \begin{cases} \dfrac{c_{\tilde{z},\ \tilde{i}} + \phi - 1}{\sum_i c_{\tilde{z},\ i} + M\phi - 1}, & \mathit{if}\ \tilde{y} = (0,0) \\ \dfrac{c_{\tilde{b},\ \tilde{i}} + \beta - 1}{\sum_i c_{\tilde{b},\ i} + M\beta - 1}, & \mathit{if}\ \tilde{y} = (1,\tilde{b}) \end{cases} \tag{A.31}$$

3. *(Ratio of cond. probs. for coupling variable)*

$$\frac{P[\mathscr{Y}|\mathscr{U},\mathscr{Z}(\tilde{z})]}{P(\mathscr{Y}^{-j}|\mathscr{U}^{-j},\ \mathscr{Z}^{-j})} = \begin{cases} \dfrac{dc_{\tilde{u},0} + \gamma - 1}{n_{\tilde{u}} + 2\gamma - 1}, & \mathit{if}\ \tilde{y} = (0,0) \\ \dfrac{dc_{\tilde{u},1} + \gamma - 1}{n_{\tilde{u}} + 2\gamma - 1} \times \dfrac{c^{\tilde{u}}_{\tilde{z},\ \tilde{b}} + \alpha - 1}{c^{1}_{\tilde{u},\ \tilde{z}} + Q\alpha - 1}, & \mathit{if}\ \tilde{y} = (1,\tilde{b}) \end{cases} \tag{A.32}$$

Proof. To derive Eqn. (A.30), we proceed exactly as in [48].

The two remaining ratios

$$\frac{P[\mathscr{I}|\mathscr{Y},\mathscr{Z}(\tilde{z})]}{P(\mathscr{I}^{-j}|\mathscr{Y}^{-j},\ \mathscr{Z}^{-j})} \text{ and } \frac{P[\mathscr{Y}|\mathscr{U},\mathscr{Z}(\tilde{z})]}{P(\mathscr{Y}^{-j}|\mathscr{U}^{-j},\ \mathscr{Z}^{-j})}$$

can be derived similarly as their counterparts in previous proof (just keep in mind that the changing component is now $\mathscr{Z}(\tilde{z})$ whereas $\mathscr{Y}$ is now fixed). □

A.3 Likelihood function

Denote parameters of BIT, i.e. the five matrices of distributions, as follows.

1. User-topic matrix of size $N \times K$: $\Theta = \left(\vartheta_u^T\right)_{u \in U}$
2. User-decision matrix of size $N \times 2$: $\Delta = \left(\delta_u^T\right)_{u \in U}$
3. Topic-brand matrix of size $K \times Q$: $\Psi = \left(\psi_z^T\right)_{z \in Z}$
4. Topic-item matrix of size $K \times M$: $\Phi = \left(\varphi_z^T\right)_{z \in Z}$
5. Brand-item matrix of size $Q \times M$: $\Omega = \left(\omega_b^T\right)_{b \in B}$

Let $\Pi = \{\Theta,\ \Delta,\ \Psi,\ \Phi,\ \Omega\}$ be all these parameters.

Given parameters Π, the likelihood of the whole adoption dataset $\mathscr{I}$ is just the product of likelihoods of all "documents" i_u i.e. we have

$$P(\mathscr{I}|\Pi) = \prod_{u} P(i_u|\Pi) \tag{A.33}$$

where the likelihood of each "document" can be estimated by the following equation

$$\begin{aligned} P(i_u|\Pi) &= \prod_{i\in I} [p(i|\Pi)]^{n_{u,i}} \qquad\qquad (n_{u,i} \text{ is the number of times } u \text{ adopted } i) \\ &= \prod_{i\in I} \left[\delta_{u,0} \times \sum_{z} \vartheta_{u,z} \cdot \phi_{z,i} + \delta_{u,1} \times \sum_{z}\sum_{b} \vartheta_{u,z} \cdot \psi_{z,b} \cdot \omega_{b,i} \right]^{n_{u,i}} \end{aligned} \tag{A.34}$$

Thus, the log likelihood function will be

$$\begin{aligned} \mathscr{L}(\Pi) &= \log P(\mathscr{I}|\Pi) \\ &= \sum_{u\in U}\sum_{i\in I} n_{u,i} \log \left[\delta_{u,0} \times \sum_{z} \vartheta_{u,z} \cdot \phi_{z,i} + \delta_{u,1} \times \sum_{z}\sum_{b} \vartheta_{u,z} \cdot \psi_{z,b} \cdot \omega_{b,i} \right] \end{aligned} \tag{A.35}$$

This explicit form of log likelihood function apparently suggests some promising things.

- If the set (or at least the number) of topics is known, we can learn parameters by solving a constrained optimization problem i.e. maximizing the log likelihood subject to probability constraints.
- Otherwise, we may try an EM approach.
- A matrix factorization approach (will be developed later) since the first and second sum suggests the products $\Theta\Phi$ and $\Theta\Psi\Omega$ resp.

A.4 Topics learned by DeBIT and LDA

This part shows topics learned by DeBIT and LDA for 4SQDB and ACMDB datasets. For each topic, we show its top 3 items with the highest probabilities in the topic's item distribution.

Table A.3: **4SQDB** – Learned topics and their top 3 venues

Topic (cuisine)	**Top venues by DeBIT**	**Top venues by LDA**
American	Swensen's, Hard Rock Cafe, Southwest Tavern	Astons Express, Swensen's, Mel's Drive-In
BBQ	Thai BBQ, Sunset Grill, Happy BBQ	Jerry's BBQ, Wong Chiew BBQ, Thai BBQ
Breakfast	Strictly Pancakes, Hatched, Sin Heng Kee Porridge,	Vanda Terrace, Coffee & Toast, Strictly Pancakes
Chinese	Yu Kee Duck House, 333 Bak Kut Teh, Ke Kou Mian	Tiong Shian Porridge, 333 Bak Kut Teh, Eminent Frog Porridge
Indian	Roti Prata House, Al-Azhar Eating, Thohirah Cafeela	Roti Prata House, Al-Azhar Eating, Casuarina Curry
Italian	Prego, Saizeriya, Oso Ristorante	PastaMania, Prego, Basilico
Japanese	Pepper Lunch Express, Nihon Mura, My Izakaya	Sakura Charcoal, The Ramen Stall, My Izakaya
Seafood	Sinma Seafood, Rasa Thai Seafood, NHC Fish-Head Steamboat	Korean Seafood, Sinma Seafood, NHC Fish-Head Steamboat

Table A.4: **ACMDB** – Learned topics and their top 3 papers. Due to space constraint, we shorten titles of the papers (readers can use paper IDs in brackets to retrieve full titles from ACMDL). To avoid repeating titles of papers found by both models, we only provide their IDs in LDA column.

Topic	Top 3 papers by DeBIT	Top 3 papers by LDA
DB+DM	BIRCH: efficient data clustering method for large DB (233324)	335372
	CURE: efficient clustering algorithm for large DB (276312)	233324
	Mining frequent patterns w/o candidate generation (335372)	276312
PO	Cache decay: reduce cache leakage power (379268)	Wattch: framework for power analysis and optimization (339657)
	Power minimization in IC design (225877)	335044
	System-level power optimization (335044)	Power aware page allocation (379007)
SE_1	The DaCapo benchmarks for java development (1167488)	236371
	Efficient path profiling (243857)	Age-based garbage collection (320425)
	Fast static analysis of C++ virtual function calls (236371)	1167488
WWW	Chord: A scalable peer-to-peer lookup service (383071)	Practical network support for IP traceback (347560)
	End-to-end Internet packet dynamics (263155)	263155
	Congestion avoidance and control (52356)	52356
System	Scale and performance in a distributed file system (35059)	35059
	Caching in the Sprite network file system (42183)	265927
	Eraser: race detector for multithreaded programs (265927)	42183
WSN	GPSR: greedy perimeter stateless routing for WN (345953)	RISA: accurate, efficient placement routability modeling (191632)
	Directed diffusion: scalable paradigm for sensor networks (345920)	586117
	Key-management scheme for distributed sensor networks (586117)	345920
DS	Unreliable failure detectors for reliable distributed systems (226647)	359563
	Time, clocks, and ordering of events in a distributed system (359563)	226647
	Web server workload characterization (233034)	Impossibility of distributed consensus with one faulty process (214121)
SE_2	Locating faulty code using failure-inducing chops (1101948)	NA
	Isolating cause-effect chains from computer programs (587053)	NA
	Evaluation of dynamic slices for fault location (1085135)	NA
Security	System design methodologies for a wireless security (514113)	Protecting privacy using the decentralized label model (363526)
	Architectural support for fast symmetric-key Crypto (379238)	Crowds: anonymity for Web transactions (290168)
	Method for obtaining digital signatures and public-key (359342)	359342
IR	NA	Self-indexing inverted files for fast text retrieval (237497)
	NA	Probabilistic relational algebra for the integration of IR and DB (239045)
	NA	NiagaraCQ: scalable continuous query system for Internet DB (335432)

Appendix B

Micro-level Diffusion Modeling with Item Interaction

To make it easy for readers, we first re-state the formulae of gradients.

B.1 Formulae of Gradients

B.1.1 Gradients for bias variables

$$\frac{\partial J}{\partial \mu} = \sum_t \sum_{u \in U} \sum_{i \in C_{u,t}} \overbrace{(\widehat{a}_{u,i,t}(\Pi) - a_{u,i,t})}^{e_{u,i,t}} \tag{B.1a}$$

$$\frac{\partial J}{\partial b_u} = \delta b_u + \sum_t \sum_{i \in C_{u,t}} e_{u,i,t} \tag{B.1b}$$

$$\frac{\partial J}{\partial b_i} = \delta b_i + \sum_t \sum_{u \in U:\, C_{u,t} \ni i} e_{u,i,t} \tag{B.1c}$$

B.1.2 Gradient for homophily variable

$$\frac{\partial J}{\partial h} = \delta h + \sum_t \sum_u p_u^T \left[S_t(u)\right] q_t^{err}(u) \tag{B.2}$$

where

$$S_t(u) = \sum_{v \in N_u} p_v (\overbrace{\sum_{j \in r_v^{k,t}} q_j}^{q_t^{(k)}(v)})^T = \sum_{v \in N_u} p_v q_t^{(k)}(v)^T \tag{B.3}$$

and

$$q_t^{err}(u) := \sum_{i \in C_{u,t}} e_{u,i,t} q_i \tag{B.4}$$

B.1.3 Gradients for user and item factors

1. (Gradient w.r.t p_u) For each given user u, we have

$$\nabla_{p_u} J = \delta p_u + \sum_t \left[M_t(u) q_t^{err}(u) + h \eta_t(u) q_t^k(u) \right] \tag{B.5}$$

   where matrix $M_t(u)$ and scalar $\eta_t(u)$ are defined as

$$M_t(u) := Id + hS_t(u) \text{ and } \eta_t(u) := \sum_{v \in N_u} p_v^T q_t^{err}(v) \tag{B.6}$$

   for directed network (e.g. Twitter), this is computed over the set $F(u)$ of followers of u.

2. (Gradient w.r.t. q_i) For each given item i, we have

$$\nabla_{q_i} J = \delta q_i + \sum_t \left[h \sum_{u \in U} \{ q_t^{err}(u) \varphi_{u,i,t}^T \} p_u + \sum_{u: C_{u,t} \ni i} e_{u,i,t} [M_t(u)]^T p_u \right] \tag{B.7}$$

   where vector $\varphi_{u,i,t} := \sum_{recent\ adopters} p_v$ is the sum of factors of neighbors who adopted i recently.

B.2 Sketch of Computations

Let x denote a parameter, gradient of objective function w.r.t x is

$$\nabla_x \mathscr{J}(.) = \frac{1}{2} [\nabla_x \mathscr{E}(.) + \delta \nabla_x \mathscr{R}(.)] \tag{B.8}$$

Recall definition of total error from Eqn. 14 in main paper

$$\mathscr{E}(\Pi) = \sum_{t=1}^{T} SE_t(\Pi) = \sum_{t=1}^{T} \sum_{u \in U} \sum_{i \in C_{u,t}} [\widehat{a}_{u,i,t}(\Pi) - a_{u,i,t}]^2$$

we have

$$\nabla_x \mathscr{E}(.) = 2 \sum_t \left(\sum_{u \in U} \sum_{i \in C_t(u)} e_{u,i,t} \nabla_x \widehat{a}_{u,i,t} \right) \tag{B.9}$$

Thus,

$$\nabla_x \mathscr{J}(.) = \sum_t \left(\sum_{u \in U} \sum_{i \in C_t(u)} e_{u,i,t} \nabla_x \widehat{a}_{u,i,t} \right) + \frac{\delta}{2} \times \nabla_x \mathscr{R}(.) \tag{B.10}$$

Since differentiating the quadratic regularizer $\mathscr{R}(.)$ is standard, we only need to focus on computing gradients $\nabla_x \widehat{a}_{u,i,t}$ of functions $\widehat{a}_{u,i,t}(\Pi)$.

Since functions $\widehat{a}_{u,i,t}(\Pi)$ are linear w.r.t variables bias $\mu, \{b_u\}_{u \in U}, \{b_i\}_{i \in I}$ and homophily h, Computing gradients for these variable is standard. Thus, we skip that part and only focus on gradients for user and item factors.

Derivation of $\nabla_{p_u} J$

Consider a given user u. To derive $\nabla_{p_u} J$, we need to know exactly which of the adoption decisions depend on p_u. Obviously, adoption decisions by u himself will depend on p_u. However, they are not the only terms depending on p_u. In fact, a adoption decision $\widehat{a}_{v,j,t}(.)$ of a neighbor v of u also depends on p_u due to the social influence of u on v (based on the formula of approximated adoption labels in TIHAD).

Hence, on computing $\nabla_{p_u} J$, we need to include gradients of terms $\widehat{a}_{u,i,t}$ as well as those of $\widehat{a}_{v,j,t}$. So we get

$$\nabla_{p_u} J = \delta p_u + \sum_t \sum_{i \in C_t(u)} \left\{ e_{u,i,t} \nabla_{p_u} \widehat{a}_{u,i,t}(\Pi) \right\} + \\ + \sum_t \sum_{v \in N(u)} \sum_{j \in C_t(v)} \left\{ e_{v,j,t} \nabla_{p_u} \widehat{a}_{v,j,t}(\Pi) \right\} \tag{B.11}$$

By Eqn. (9), $\widehat{a}_{u,i,t}$ has the following form

$$\begin{aligned} \widehat{a}_{u,i,t}(\Pi) &= bias(u,i) + p_u^T \overbrace{[Id + hS_t(u)]}^{M_t(u)} q_i \\ &= bias(u,i) + p_u^T \left[M_t(u) q_i \right] \end{aligned}$$

which is linear w.r.t. p_u. Thus,

$$\nabla_{p_u} \widehat{a}_{u,i,t}(\Pi) = M_t(u) q_i \tag{B.12}$$

For each estimation $\widehat{a}_{v,j,t}(\Pi)$ for a neighbor v of u, p_u only appears once in the linear term $hp_v^T p_u \left[q_t^k(u)^T q_j \right]$. Thus, we get

$$\nabla_{p_u} \widehat{a}_{v,j,t}(\Pi) = h \left[q_j^T q_t^k(u) \right] p_v \tag{B.13}$$

Plug Eqns. B.12, B.13 into Eqn. B.11 and perform some basic operations we finish derivation of $\nabla_{p_u} J$ and obtain desired Eqn. B.5.

Derivation of $\nabla_{q_i} J$

Consider a given item i. To derive $\nabla_{q_i} J$ we need to know exactly which adoption decisions actually contain q_i. Obviously q_i is contained in adoption decisions $\widehat{a}_{u,i,t}(.)$ if $i \in C_t(u)$. However, these are not the only adoption decisions which involve q_i. In fact, due to interaction among items, adoption decisions $\widehat{a}_{u,j,t}(.)$ for another item j can also contain q_i. Let us look at the adoption decision carefully.

$$\widehat{a}_{u,j,t}(.) = bias\ + p_u^T \left[I + h \sum_{v \in N(u)} p_v\, q_t^k(v)^T \right] q_j$$

We can see that q_i can appear in the term $q_t^k(v)$. This happens if $i \in k_t(v)$ (i in top-k recent items adopted by v). Thus, if there exists neighbor v of u such that v adopted i recently, adoption decision $\widehat{a}_{u,j,t}(.)$ will involve q_i. In summary there are two types of adoption decisions which involve q_i. They are summarized in the following remark.

Remark 1. *For a given item i, adoption decisions which depend on q_i can only be one of the two following types.*

- *$\widehat{a}_{u,i,t}(.)$, if this term is valid i.e. when i belongs to the set $C_t(u)$.*
- *$\widehat{a}_{u,j,t}(.)$, for items j's different from i. This can only happen if there exist neighbors of u who adopted i recently. Denote the set of such neighbors as $N^i_{\simeq t}(u)$.*

This remark suggests that gradients should be computed differently depending on whether i belongs to $C_t(u)$ or not. This will lead to two scenarios in Lemma 8. To state the result, we first need a quantity defined as follows.

Definition 17. *For each user u such that $N^i_{\simeq t}(u) \neq \emptyset$, i.e. there exist one or more neighbors of u who adopted i before time t, we define*

$$\phi_{u,i,t} := \sum_{v \in N^i_{\simeq t}(u)} p_v \tag{B.14}$$

Given this notation, we now can state the main lemma used in computing $\nabla_{q_i} J$.

Lemma 8. *Consider a user u, depending on whether i belongs to $C_t(u)$, we have two following scenarios.*

1. *If $i \notin C_t(u)$ then we have*

$$\sum_{j \in C_t(u)} e_{u,j,t} \nabla_{q_i} \left[\widehat{a}_{u,j,t}(.) \right] = h \times \left[q_t^{err}(u) \phi^T_{u,i,t} \right] p_u \tag{B.15}$$

2. *If $i \in C_t(u)$ then we have*

$$\sum_{j \in C_t(u)} e_{u,j,t} \nabla_{q_i} \left[\widehat{a}_{u,j,t}(.) \right] = h \times \left\{ q_t^{err}(u) \phi_{u,i,t}^T \right\} p_u + e_{u,i,t} \left[M(u,t) \right]^T p_u \tag{B.16}$$

Proving this main lemma requires two following additional lemmas.

Lemma 9. *Given a user u and an item $j \neq i$, gradient w.r.t. variable q_i of term $\widehat{a}_{u,j,t}(.)$ can be computed as follows.*

$$\nabla_{q_i} \left[\widehat{a}_{u,j,t}(.) \right] = h \times \left(q_j \phi_{u,i,t}^T p_u \right) \tag{B.17}$$

Lemma 10. *Consider a user u who has not adopted i before t. Then the term $\widehat{a}_{u,i,t}(.)$ is valid and its gradient w.r.t. q_i is given by*

$$\nabla_{q_i} \left[\widehat{a}_{u,i,t}(.) \right] = \left[h q_i \phi_{u,i,t}^T + M_t(u)^T \right] p_u \tag{B.18}$$

We provide proofs of the lemmas in the following sections.

B.3 Proof of Lemma 8

Proof. Why two scenarios? The reason can be explained as follows. If $i \notin C_t(u)$, the term $\widehat{a}_{u,i,t}(.)$ is invalid and thus not present in objective function. On the contrary, if $i \in C_t(u)$, the term $\widehat{a}_{u,i,t}(.)$ is now valid and thus must be taken into account when we compute gradients of such user.

By this explanation, we now see that, for users u whose $C_t(u)$ does not contain i, we only need gradients $\nabla_{q_i} \left[\widehat{a}_{u,j,t}(.) \right]$ for items $j \neq i$. These are provided in Lemma 9.

Now, for users u whose $C_t(u)$ contains i we have two kinds of gradients

1. Gradients of $\widehat{a}_{u,j,t}(.)$ where $j \neq i$.

2. Gradient of $\widehat{a}_{u,i,t}(.)$.

The first kind is exactly the same as before. Thus, they create the first term in Eqn. B.16. The second kind of gradient is provided in Lemma 10. □

B.4 Proof of Lemma 9

Proof. For a user u and for each item $j \neq i$, we can split $\widehat{a}_{u,j,t}(.)$ into two following parts, one will not depend on q_i while the other depend on the term.

$$\begin{aligned}
\widehat{a}_{u,j,t}(.) &= \overbrace{bias\ + p_u^T q_j + hp_u^T \left[\sum_{v \in N(u) \setminus N^i_{\simeq t}(u)} p_v\, q_t^k(v)^T \right] q_j}^{part_1:\ not\ depend\ on\ q_i} \\
&\quad + \overbrace{hp_u^T \left[\sum_{v \in N^i_{\simeq t}(u)} p_v\, q_t^k(v)^T \right] q_j}^{part_2:\ depend\ on\ q_i} \\
&= part_1\ + part_2
\end{aligned} \tag{B.19}$$

Thanks to this split, on computing gradient $\nabla_{q_i}\left[\widehat{a}_{u,j,t}(.)\right]$, the first part will vanish. Thus, we have

$$\nabla_{q_i}\left[\widehat{a}_{u,j,t}(.)\right] = \nabla_{q_i}(part_2) \tag{B.20}$$

The second part, $part_2$, is a sum of terms which correspond to neighbors $v \in N^i_{\simeq t}(u)$. Each such term has the following form (see Eqn. B.3 for definition of $q_t^k(v)$)

$$hp_u^T\left[p_v\, q_t^k(v)^T\right] q_j = h\left(p_u^T p_v\right) \times \left[q_i + \sum_{i' \in I_t(v) \setminus i} q_{i'} \right]^T q_j \tag{B.21}$$

We can see that the last sum on the RHS does not depend on q_i. Thus, gradient of the LHS is gradient of the first term. This term is linear w.r.t. q_i and we get

$$\nabla_{q_i}\left(hp_u^T\left[p_v\, q_t^k(v)^T\right] q_j\right) = h\left(p_u^T p_v\right) q_j = h\left(q_j p_v^T\right) p_u$$

Thus, we obtain

$$\nabla_{q_i}(part_2) = hq_j \left[\sum_{v \in N^i_{\simeq t}(u)} p_v \right]^T p_u = h \times \left(q_j \phi^T_{u,i,t} p_u \right) \tag{B.22}$$

the last equality comes from Definition B.14 of $\phi_{u,i,t}$.

Combining Eqns. B.20 and B.22, we finally obtain

$$\nabla_{q_i}\left[\widehat{a}_{u,j,t}(.)\right] = h \times \left(q_j \phi^T_{u,i,t} p_u \right)$$

which finishes the proof. □

B.5 Proof of Lemma 10

Proof. Using a split similar as in Lemma 9, we now can divide $\widehat{a}_{u,i,t}(.)$ into three terms: bias (which is constant w.r.t. q_i), a linear and a quadratic function of q_i respectively.

$$\widehat{a}_{u,i,t}(.) = bias + \overbrace{p_u^T \left[Id + h \sum_{\substack{v \in N(u) \\ v \notin N^i_{\simeq t}(u)}} p_v \, q^k_t(v)^T \right] q_i}^{part_1 : linear\ function\ of\ q_i}$$
$$+ \overbrace{hp_u^T \left[\sum_{v \in N^i_{\simeq t}(u)} p_v \, q^k_t(v)^T \right] q_i}^{part_2 : quadratic\ function\ of\ q_i} \tag{B.23}$$

Recall that the average vector $q^k_t(v)$ involve all vectors q_j of items j which were adopted by v before time t. Thus, for a user $v \notin N^i_{\simeq t}(u)$, who has not adopted i before time t, his average vector $q^k_t(v)$ does not involve q_i. This is the reason which makes $part_1$ linear w.r.t. q_i. On the contrary, for $v \in N^i_{\simeq t}(u)$, his vector $q^k_t(v)$ does contain q_i, which makes $part_2$ become a quadratic function of q_i.

Due to the split in Eqn. B.23, we have

$$\nabla_{q_i}\left[\widehat{a}_{u,i,t}(.)\right] = \nabla_{q_i}\left(part_1\right) + \nabla_{q_i}\left(part_2\right) \tag{B.24}$$

Rewriting the linear part as $p_u^T A q_i$ where

$$A = Id + h \sum_{\substack{v \in N(u) \\ v \notin N^i_{\simeq t}(u)}} p_v q_t^k(v)^T$$

we get

$$\nabla_{q_i}\left(part_1\right) = A^T p_u = \left[Id + h \sum_{\substack{v \in N(u) \\ v \notin N^i_{\simeq t}(u)}} q_t^k(v) p_v^T \right] p_u \tag{B.25}$$

Let us proceed to the quadratic part. Each $v \in N^i_{\simeq t}(u)$ contributes the following term to the quadratic part.

$$\begin{aligned} h \times (p_u^T p_v) q_t^k(v)^T q_i &= h(p_u^T p_v)\left(q_i + \sum_{j \neq i} q_j\right)^T q_i \\ &= h(p_u^T p_v)\left(\underbrace{q_i^T q_i}_{quadratic} + \underbrace{\left[\sum_{j \neq i} q_j\right]^T q_i}_{linear}\right) \end{aligned} \tag{B.26}$$

Thus, for each $v \in N^i_{\simeq t}(u)$, gradient of the corresponding term is

$$\begin{aligned} \nabla_{q_i}(LHS) &= h \times (p_u^T p_v)\left[2q_i + \sum_{j \neq i} q_j\right] \\ &= h \times (p_u^T p_v)\left(q_i + q_t^k(v)\right) \end{aligned} \tag{B.27}$$

Summing over $v \in N^i_{\simeq t}(u)$, we obtain the gradient of the quadratic term $part_2$ as follows.

$$\begin{aligned}
\nabla_{q_i}(part_2) &= h\left[\sum_{v\in N^i_{\simeq t}(u)} q^k_t(v)p^T_v\right]p_u + h\left[q_i \underbrace{\sum_{v\in N^i_{\simeq t}(u)} p^T_v}_{\phi^T_{u,i,t}}\right]p_u \\
&= h\left[\sum_{v\in N^i_{\simeq t}(u)} q^k_t(v)p^T_v\right]p_u + h\left[q_i\phi^T_{u,i,t}\right]p_u \qquad \text{(B.28)}
\end{aligned}$$

In summary, from Eqns. B.25 and B.28, we have

$$\begin{aligned}
\nabla_{q_i}(part_1) &= \left[Id + h\sum_{\substack{v\in N(u)\\ v\notin N^i_{\simeq t}(u)}} q^k_t(v)p^T_v\right]p_u \\
\nabla_{q_i}(part_2) &= h\left[\sum_{v\in N^i_{\simeq t}(u)} q^k_t(v)p^T_v\right]p_u + h\left[q_i\phi^T_{u,i,t}p_u\right]
\end{aligned}$$

Adding the two equations together, we get

$$\begin{aligned}
\nabla_{q_i}\left[\widehat{a}_{u,i,t}(.)\right] &= \nabla_{q_i}(part_1) + \nabla_{q_i}(part_2) \\
&= \left[\underbrace{Id + h\sum_{v\in N(u)} p_v q^k_t(v)^T}_{M_t(u)}\right]^T p_u + h\left[q_i\phi^T_{u,i,t}p_u\right] \\
&= \left[M_t(u)\right]^T p_u + h\left[q_i\phi^T_{u,i,t}p_u\right] \\
&= \left[hq_i\phi^T_{u,i,t} + M_t(u)^T\right]p_u \qquad \text{(B.29)}
\end{aligned}$$

which finishes our proof. □

Bibliography

[1] David A Aaker. The value of brand equity. *Journal of business strategy*, 13 (4):27–32, 1992.

[2] David A Aaker. Brand extensions: the good, the bad and the ugly. *Sloan Management Review*, 31(4), 2012.

[3] acm. ACM Digital Library. `http://dl.acm.org/`, 2011.

[4] Ryan Prescott Adams, George E Dahl, and Iain Murray. Incorporating side information in probabilistic matrix factorization with gaussian processes. *arXiv preprint arXiv:1003.4944*, 2010.

[5] Edo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. In *NIPS*, pages 33–40, 2009.

[6] Tim Ambler. Do brands benefit consumers? *International Journal of Advertising*, 16(3):167–198, 1997.

[7] Ramnath Balasubramanyan and William W Cohen. Block-lda: Jointly modeling entity-annotated text and entity-entity links. In *SDM*, volume 11, pages 450–461, 2011.

[8] Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. Matrix factorization techniques for context aware recommendation. In *Rec.Sys*, pages 301–304, 2011.

[9] Punam Bedi, Harmeet Kaur, and Sudeep Marwaha. Trust based recommender system for semantic web. *International Joint Conference on Artificial Intelligence (IJCAI)*, 7:2677–2682, 2007.

[10] Ana Belén del Río, Rodolfo Vazquez, and Victor Iglesias. The effects of brand associations on consumer response. *Journal of Consumer Marketing*, 18(5):410–425, 2001.

[11] Alex Beutel, B Aditya Prakash, Roni Rosenfeld, and Christos Faloutsos. Interacting viruses in networks: can both survive? In *SIGKDD*, pages 426–434. ACM, 2012.

[12] David M Blei and John D Lafferty. Dynamic topic models. In *ICML*, pages 113–120, 2006.

[13] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research (JMLR)*, 3, 2003.

[14] Joseph K Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin. Parallel coordinate descent for l1-regularized loss minimization. *arXiv preprint arXiv:1105.5379*, 2011.

[15] Marilynn B Brewer. The social self: On being the same and different at the same time. *Personality and Social Psychology Bulletin*, 17(5):475–482, 1991.

[16] George Casella and Edward I George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[17] Arjun Chaudhuri and Morris B Holbrook. The chain of effects from brand trust and brand affect to brand performance: the role of brand loyalty. *The Journal of Marketing*, 65(2):81–93, 2001.

[18] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for

prevalent viral marketing in large-scale social networks. In *SIGKDD*, pages 1029–1038, 2010.

[19] Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*, pages 88–97. IEEE, 2010.

[20] Nicholas A Christakis and James H Fowler. The spread of obesity in a large social network over 32 years. *New England journal of medicine*, 357(4): 370–379, 2007.

[21] Freddy Chong Tat Chua, Hady Wirawan Lauw, and Ee-Peng Lim. Predicting item adoption using social correlation. In *SDM*, volume 11, pages 367–378, 2011.

[22] Freddy Chong Tat Chua, Hady W Lauw, and Ee-Peng Lim. Generative models for item adoptions using social correlation. *TKDE*, 25(9):2036–2048, 2013.

[23] Freddy Chong Tat Chua, Richard J Oentaryo, and Ee-Peng Lim. Modeling temporal adoptions using dynamic matrix factorization. In *ICDM*, pages 91–100, 2013.

[24] David Crandall, Dan Cosley, Daniel Huttenlocher, Jon Kleinberg, and Siddharth Suri. Feedback effects between similarity and social influence in online communities. In *SIGKDD*, pages 160–168, 2008.

[25] Henggang Cui, James Cipar, Qirong Ho, Jin Kyu Kim, Seunghak Lee, Abhimanu Kumar, Jinliang Wei, Wei Dai, Gregory R Ganger, Phillip B Gibbons, et al. Exploiting bounded staleness to speed up big data analytics. In *USENIX Annual Technical Conference, USENIX ATC*, volume 14, pages 37–48, 2014.

[26] Ali Daud, Juanzi Li, Lizhu Zhou, and Faqir Muhammad. Exploiting temporal authors interests via temporal-author-topic modeling. In *Advanced Data Mining and Applications*, pages 435–443. Springer, 2009.

[27] Munmun De Choudhury, Hari Sundaram, Ajita John, Doree Duncan Seligmann, and Aisling Kelliher. ”birds of a feather”: Does user homophily impact information diffusion in social media? *arXiv preprint arXiv:1006.1702*, 2010.

[28] Julien Delporte, Alexandros Karatzoglou, Tomasz Matuszczyk, and Stéphane Canu. Socially enabled preference learning from implicit feedback data. In *ECMLPKDD*, pages 145–160. 2013.

[29] Arthur P Dempster, Nan M Laird, and Donald 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.

[30] Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. Finding bursty topics from microblogs. In *Association for Computational Linguistics*, pages 536–544, 2012.

[31] Alan S Dick and Kunal Basu. Customer loyalty: toward an integrated conceptual framework. *Journal of the Academy of Marketing Science*, 22(2): 99–113, 1994.

[32] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *SIGKDD*, pages 126–135, 2006.

[33] Chris HQ Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SDM*, pages 606–610, 2005.

[34] Dominik Maria Endres and Johannes E Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory (TOIS)*, 49, 2003.

[35] Tlin Erdem and Michael P. Keane. Decision-making under uncertainty: Capturing dynamic brand choice processes in turbulent consumer goods markets. *Marketing Science*, 15(1):pp. 1–20, 1996.

[36] Tülin Erdem and Joffre Swait. Brand equity as a signaling phenomenon. *Journal of Consumer Psychology*, 7(2):131–157, 1998.

[37] Tülin Erdem, Joffre Swait, Susan Broniarczyk, Dipankar Chakravarti, Jean-Noel Kapferer, Michael Keane, John Roberts, Jan-Benedict EM Steenkamp, and Florian Zettelmeyer. Brand equity, consumer learning and choice. *Marketing Letters*, 10(3):301–318, 1999.

[38] Tülin Erdem, Ying Zhao, and Ana Valenzuela. Performance of store brands: A cross-country analysis of consumer store-brand preferences, perceptions, and risk. *Journal of Marketing Research*, 41(1):86–100, 2004.

[39] Tom Fawcett. Roc graphs: Notes and practical considerations for researchers. *Machine Learning*, 31:1–38, 2004.

[40] Noah E Friedkin. *A structural theory of social influence*, volume 13. Cambridge University Press, 2006.

[41] Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410), 1990.

[42] Sahin Cem Geyik, Ali Dasdan, and Kuang-Chih Lee. User clustering in online advertising via topic models. *arXiv preprint arXiv:1501.06595*, 2015.

[43] Benjamin Golub and Matthew O Jackson. How homophily affects the speed of learning and best-response dynamics. *The Quarterly Journal of Economics*, 127(3):1287–1338, 2012.

[44] Timothy R Graeff. Using promotional messages to manage the effects of brand and self-image on brand evaluations. *Journal of Consumer Marketing*, 13(3):4–18, 1996.

[45] Mark Granovetter. Threshold models of collective behavior. *American journal of sociology*, pages 1420–1443, 1978.

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

[47] Young Jee Han, Joseph C Nunes, and Xavier Drèze. Signaling status with luxury goods: The role of brand prominence. *Journal of Marketing*, 74(4): 15–30, 2010.

[48] Gregor Heinrich. Parameter estimation for text analysis. `http://www.arbylon.net/publications/text-est.pdf`, 2005.

[49] Qirong Ho, James Cipar, Henggang Cui, Seunghak Lee, Jin Kyu Kim, Phillip B Gibbons, Garth A Gibson, Greg Ganger, and Eric P Xing. More effective distributed ML via a stale synchronous parallel parameter server. In *NIPS*, pages 1223–1231, 2013.

[50] Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for latent dirichlet allocation. In *NIPS*, 2010.

[51] Thomas Hofmann. Latent semantic models for collaborative filtering. *TOIS*, 22(1):89–115, 2004.

[52] Margaret K Hogg, Alastair J Cox, and Kathy Keeling. The impact of self-monitoring on image congruence and product/brand evaluation. *European Journal of Marketing*, 34(5/6):641–667, 2000.

[53] John A Howard and Jagdish N Sheth. *The theory of buyer behavior*. Wiley New York, 1969.

[54] Patrik O Hoyer. Non-negative sparse coding. In *Neural Networks for Signal Processing*, pages 557–565, 2002.

[55] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[56] Lun-ping Hung. A personalized recommendation system based on product taxonomy for one-to-one marketing online. *Expert systems with applications*, 29(2):383–392, 2005.

[57] hungrygowhere. `http://www.hungrygowhere.com/`, 2015.

[58] Jacob Jacoby and Robert W Chestnut. *Brand loyalty: Measurement and management*. Wiley New York, 1978.

[59] Niklas Jakob, Stefan Hagen Weber, Mark Christoph Müller, and Iryna Gurevych. Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 57–64. ACM, 2009.

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

[61] Carl T Kelley. *Iterative methods for optimization*, volume 18. Siam, 1999.

[62] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, pages 137–146, 2003.

[63] Jeehong Kim and Wonchang Hur. Diffusion of competing innovations in influence networks. *Journal of Economic Interaction and Coordination*, 8 (1):109–124, 2013.

[64] Jin Kyu Kim, Qirong Ho, Seunghak Lee, Xun Zheng, Wei Dai, Garth A Gibson, and Eric P Xing. Strads: a distributed framework for scheduled model parallel machine learning. In *EuroSys*, page 5, 2016.

[65] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. Solving the contamination minimization problem on networks for the linear threshold model. In

PRICAI 2008: Trends in Artificial Intelligence, volume 5351, pages 977–984. 2008.

[66] Noam Koenigstein, Gideon Dror, and Yehuda Koren. Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. In *Rec.Sys*, pages 165–172, 2011.

[67] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*, pages 426–434, 2008.

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

[69] Miroslav Kubat, Stan Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *ICML*, volume 97, pages 179–186, 1997.

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

[71] DD Lee and HS Seung. Algorithms for non-negative matrix factorization. *NIPS*, 2001.

[72] Seunghak Lee, Jin Kyu Kim, Xun Zheng, Qirong Ho, Garth A Gibson, and Eric P Xing. On model parallelization and scheduling strategies for distributed machine learning. In *Advances in neural information processing systems*, pages 2834–2842, 2014.

[73] Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 583–598, 2014.

[74] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10):2756–2779, 2007.

[75] Jianhua Lin. Divergence measures based on the shannon entropy. *TOIS*, 37 (1):145–151, 1991.

[76] Shuyang Lin, Qingbo Hu, Fengjiao Wang, and Philip S Yu. Steering information diffusion dynamically against user attention limitation. *ICDM*, 2014.

[77] Erik Linstead, Paul Rigor, Sushil Bajracharya, Cristina Lopes, and Pierre Baldi. Mining eclipse developer contributions via author-topic models. In *ICSE Mining Software Repositories workshops*, pages 30–33, 2007.

[78] Weixiang Liu, Nanning Zheng, and Xiaofeng Lu. Non-negative matrix factorization for visual coding. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–293, 2003.

[79] Mary M Long and Leon G Schiffman. Consumption values and relationships: segmenting the market for frequency programs. *Journal of Consumer Marketing*, 17(3):214–232, 2000.

[80] Yucheng Low, Danny Bickson, Joseph Gonzalez, Carlos Guestrin, Aapo Kyrola, and Joseph M Hellerstein. Distributed graphlab: a framework for machine learning and data mining in the cloud. *Proceedings of the VLDB Endowment*, 5(8):716–727, 2012.

[81] Duc M. Luu. Topics and their corresponding top papers learned by BITM. `http://goo.gl/nuMnIw`, 2013.

[82] Duc M. Luu. Derivations of Gibbs sampling for eBIT. `https://goo.gl/Jl97X6`, 2015.

[83] Minh-Duc Luu and Ee-Peng Lim. Latent factors meet homophily in diffusion

modelling. In *Machine Learning and Knowledge Discovery in Databases*, pages 701–718. Springer, 2015.

[84] Minh Duc Luu, Ee-Peng Lim, and Freddy Chong Tat Chua. On modeling brand preferences in item adoptions. In *ICWSM*, 2014.

[85] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, pages 931–940, 2008.

[86] Hao Ma, Irwin King, and Michael R Lyu. Learning to recommend with social trust ensemble. In *SIGIR*, pages 203–210, 2009.

[87] Hao Ma, Michael R Lyu, and Irwin King. Learning to recommend with trust and distrust relationships. In *Rec.Sys*, pages 189–196, 2009.

[88] Hao Ma, Irwin King, and Michael R Lyu. Learning to recommend with explicit and implicit social relations. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):29, 2011.

[89] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *WSDM*, pages 287–296, 2011.

[90] Hao Ma, Tom Chao Zhou, Michael R Lyu, and Irwin King. Improving recommender systems by incorporating social contextual information. *ACM Transactions on Information Systems (TOIS)*, 29(2):9, 2011.

[91] Paolo Massa and Paolo Avesani. Trust-aware recommender systems. In *RecSys*, pages 17–24. ACM, 2007.

[92] Robert M May and Warren J Leonard. Nonlinear aspects of competition between three species. *SIAM Journal on Applied Mathematics*, 29(2):243–253, 1975.

[93] Andrew McCallum, Andres Corrada-Emmanuel, and Xuerui Wang. Topic and role discovery in social networks. *Computer Science Department Faculty Publication Series*, page 3, 2005.

[94] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, pages 415–444, 2001.

[95] Qiaozhu Mei, Deng Cai, Duo Zhang, and ChengXiang Zhai. Topic modeling with network regularization. In *WWW*, pages 101–110, 2008.

[96] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2007.

[97] Chris A Myers. Managing brand equity: a look at the impact of attributes. *Journal of Product & Brand Management*, 12(1):39–51, 2003.

[98] Seth A Myers and Jure Leskovec. Clash of the contagions: Cooperation and competition in information diffusion. In *ICDM*, pages 539–548, 2012.

[99] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed algorithms for topic models. *JMLR*, 10:1801–1828, 2009.

[100] István Pilászy, Dávid Zibriczky, and Domonkos Tikk. Fast als-based matrix factorization for explicit and implicit feedback datasets. In *Rec.Sys*, pages 71–78, 2010.

[101] Ian Porteous, Arthur U Asuncion, and Max Welling. Bayesian matrix factorization with side information and dirichlet process mixtures. In *AAAI*, 2010.

[102] B Aditya Prakash, Alex Beutel, Roni Rosenfeld, and Christos Faloutsos. Winner takes all: competing viruses or ideas on fair-play networks. In *WWW*, pages 1037–1046. ACM, 2012.

[103] Xiang Qi, Yu Huang, Ziyan Chen, Xiaoyan Liu, Jing Tian, Tinglei Huang, and Hongqi Wang. Burst-lda: A new topic model for detecting bursty topics from stream text. *Journal of Electronics (China)*, 31(6):565–575, 2014.

[104] Xueming Qian, He Feng, Guoshuai Zhao, and Tao Mei. Personalized recommendation combining user interest and social circle. *TKDE*, 26(7):1763–1777, 2014.

[105] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*, pages 693–701, 2011.

[106] Richard A Redner and Homer F Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM review*, 26(2):195–239, 1984.

[107] Steffen Rendle and Lars Schmidt-Thieme. Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *Rec.Sys*, pages 251–258, 2008.

[108] EM Rogers. *Diffusion of innovations.* New York (USA), Free Press, 1983.

[109] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *UAI*, pages 487–494, 2004.

[110] Michal Rosen-Zvi, Chaitanya Chemudugunta, Thomas Griffiths, Padhraic Smyth, and Mark Steyvers. Learning author-topic models from text corpora. *ACM Transactions on Information Systems (TOIS)*, 28(1):4–41, 2010.

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

[112] TC Schelling. Micromotives and macrobehavior. 1978.

[113] Farial Shahnaz, Michael W Berry, V Paul Pauca, and Robert J Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386, 2006.

[114] Hanhuai Shan, Jens Kattge, Peter Reich, Arindam Banerjee, Franziska Schrodt, and Markus Reichstein. Gap filling in the plant kingdom—trait prediction using hierarchical probabilistic matrix factorization. *arXiv preprint arXiv:1206.6439*, 2012.

[115] Yelong Shen and Ruoming Jin. Learning personal + social latent factor model for social recommendation. In *SIGKDD*, pages 1303–1311, 2012.

[116] Michael D. Smith and Erik Brynjolfsson. Consumer decision-making at an internet shopbot: Brand still matters. *The Journal of Industrial Economics*, 49(4):541–558, 2001.

[117] Alexander Smola and Shravan Narayanamurthy. An architecture for parallel topic models. *Proceedings of the VLDB Endowment*, 3(1-2):703–710, 2010.

[118] Michael R Solomon. The value of status and the status of value. *Consumer Value: a Framework for Analysis and Research*, pages 63–84, 1999.

[119] Mark Steyvers, Padhraic Smyth, Michal Rosen-Zvi, and Thomas Griffiths. Probabilistic author-topic models for information discovery. In *SIGKDD*, pages 306–315, 2004.

[120] Dong Liang Su, Zhi Ming Cui, Jian Wu, and Peng Peng Zhao. Pre-filling collaborative filtering algorithm based on matrix factorization. *Trans Tech Publ*, 411:2223–2228, 2013.

[121] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[122] Songgao Tu and Chaojun Lu. Topic-based user segmentation for online advertising with latent dirichlet allocation. In *Advanced Data Mining and Applications*, pages 259–269. 2010.

[123] David Ubilava, Kenneth A Foster, Jayson L Lusk, and Tomas Nilsson. Differences in consumer preferences when facing branded versus non-branded choices. *Journal of Consumer Behaviour*, 10(2):61–70, 2011.

[124] Franck Vigneron and Lester W Johnson. A review and a conceptual framework of prestige-seeking consumer behavior. *Academy of Marketing Science Review*, 1999:1, 1999.

[125] Yuka Wakita, Kenta Oku, Hung-Hsuan Huang, and Kyoji Kawagoe. A fashion-brand recommender system using brand association rules and features. In *IIAI 4th International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 719–720, 2015.

[126] Dingding Wang, Tao Li, Shenghuo Zhu, and Chris Ding. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *SIGIR*, pages 307–314, 2008.

[127] Xiang Wang, Kai Zhang, Xiaoming Jin, and Dou Shen. Mining common topics from multiple asynchronous text streams. In *WSDM*, pages 192–201, 2009.

[128] Xuanhui Wang, ChengXiang Zhai, Xiao Hu, and Richard Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *SIGKDD*, pages 784–793, 2007.

[129] L. Weng, A. Flammini, A. Vespignani, and F. Menczer. Competition among memes in a world with limited attention. *Scientific Reports*, 2012.

[130] Xiaohui Wu, Jun Yan, Ning Liu, Shuicheng Yan, Ying Chen, and Zheng Chen. Probabilistic latent semantic user segmentation for behavioral targeted

advertising. In *International Workshop on Data Mining and Audience Intelligence for Advertising*, pages 10–17. ACM, 2009.

[131] Wei Xie, Feida Zhu, Jing Jiang, Ee-Peng Lim, and Ke Wang. Topicsketch: Real-time bursty topic detection from twitter. In *ICDM*, pages 837–846, 2013.

[132] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *SIGIR*, pages 267–273, 2003.

[133] Zhiheng Xu, Rong Lu, Liang Xiang, and Qing Yang. Discovering user interest on twitter with a modified author-topic model. In *International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 1, pages 422–429, 2011.

[134] De-Nian Yang, Wang-Chien Lee, Nai-Hui Chia, Mao Ye, and Hui-Ju Hung. On bundle configuration for viral marketing in social networks. In *CIKM*, pages 2234–2238, 2012.

[135] James A Yorke and William N Anderson Jr. Predator-prey patterns. *PNAS*, 70(7):2069, 1973.

[136] Jinhui Yuan, Fei Gao, Qirong Ho, Wei Dai, Jinliang Wei, Xun Zheng, Eric Po Xing, Tie-Yan Liu, and Wei-Ying Ma. Lightlda: Big topic models on modest computer clusters. In *WWW*, pages 1351–1361, 2015.

[137] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *SDM*, volume 6, pages 548–552, 2006.

[138] Yongzheng Zhang and Marco Pennacchiotti. Recommending branded products from social media. In *RecSys*, pages 77–84, 2013.

[139] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. Parallelized stochastic gradient descent. In *NIPS*, pages 2595–2603, 2010.