PREDICTIVE MODELS FOR ONLINE HUMAN ACTIVITIES

A Thesis
Presented to
The Academic Faculty

by

Shuang-Hong Yang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Computer Science

School of Computer Science, College of Computing
Georgia Institute of Technology
May 2012

PREDICTIVE MODELS FOR ONLINE HUMAN ACTIVITIES

Approved by:

Professor Hongyuan Zha, Advisor
College of Computing
Georgia Institute of Technology

Professor Ming Yuan
School of Industrial and System Engineering
Georgia Institute of Technology

Professor Guy Lebanon
College of Computing
Georgia Institute of Technology

Professor Eugene Agichtein
Mathematics & Computer Science Department
Emory University

Professor Le Song
College of Computing
Georgia Institute of Technology

Date Approved: March 27, 2012

To my beloved

Juan and Harris.

ACKNOWLEDGEMENTS

First, I want to thank my advisor Professor Hongyuan Zha. It is you who brought me to this exciting area, inspired me with sound advices and encouraged me to attack real-world challenges. This dissertation would not have been made possible without your inspiration, advice and support.

I would like to thank my co-mentors, Bo Long and Alexander Smola from Yahoo! Labs. It was the most pleasant and memorable experience in my life working with you. Thank you for the opportunity you opened to me and for all the great advices you provided to both my research and career endeavor.

I am grateful to my teammates Ke Zhou, Steven Crain, Thomas Perry and Jiang Bian, and to my colleagues in the DAS lab, Fuxin Li, Mingxuan Sun, Liangda Li, Josh Dillion and many others. PhD life has been so much colorful and enjoyable because of you. Thanks for being with me.

I would also like to express my gratitude to my thesis committee members, Professor Guy Lebanon, Professor Le Song, Professor Ming Yuan and Professor Eugene Agichtein. Many thanks for your insightful advices and constructive feedbacks to help improve the quality of the thesis.

Finally and most importantly, I thank my beloved wife, Juan, our baby boy, Harris, and our parents. Thank you for your endless love, patience, support and encouragement. To you I dedicate this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The availability and scale of user generated data in online systems raises tremendous challenges and opportunities to analytic study of human activities. Effective modeling of online human activities is not only fundamental to the understanding of human behavior, but also important to the online industry. This thesis focuses on developing models and algorithms to predict human activities in online systems and to improve the algorithmic design of personalized/socialized systems (e.g., recommendation, advertising, Web search systems). We are particularly interested in three types of online user activities, i.e., decision making, social interactions and user-generated contents. Centered around these activities, the thesis focuses on three challenging topics:

1. Behavior prediction, i.e., predicting users' online decisions. We present *Collaborative-Competitive Filtering*, a novel game-theoretic framework for predicting users' online decision making behavior and leverage the knowledge to optimize the design of online systems (e.g., recommendation systems) in respect of certain strategic goals (e.g., sales revenue, consumption diversity).

2. Social contagion, i.e., modeling the interplay between social interactions and individual behavior of decision making. We establish the joint *Friendship-Interest Propagation* model and the *Behavior-Relation Interplay* model, a series of statistical approaches to characterize the behavior of individual user's decision making, the interactions among socially connected users, and the interplay between these two activities. These techniques are demonstrated by applications to social behavior targeting.

3. Content mining, i.e., understanding user generated contents. We propose the *Topic-Adapted Latent Dirichlet Allocation* (τLDA) model, a probabilistic model for identifying a user's hidden cognitive aspects (e.g., knowledgability) from the texts created by the user. The model is successfully applied to address the challenge of "language gap" in medical information retrieval.

Keywords: Predictive models, online human activities, user-generated data, behavior prediction, social ties, collaborative competitive filtering, social contagion, behavior-relation interplay, content mining, user cognitive aspects, language gap

CHAPTER I

INTRODUCTION

One of the central topics in many disciplines of social science (e.g., economics, psychology, sociology) is to understand human activities, e.g., people's decision making behaviors, social interactions and natural languages. The emergence of online systems (e.g., online merchants, social media, discussion forums) has contributed to this study many new promises. In particular, human activity data are generated spontaneously by real people and available at extremely large scales we have never seen before. Moreover, compared to the offline physical world, online systems have the unbeatable convenience in measuring and intervening the decision environment. The availability and scale of human activities and the convenience of manipulation in online systems raises tremendous opportunities at the interface of social science and computer science: on the one hand, it allows social scientist to leverage computational power to test justifiable hypotheses and draw significant insights from large scale human behavioral phenomena; and on the other hand, it motivates computer scientists to develop powerful tools to help make sense of the big data and facilitate knowledge discovery.

The focus of this dissertation is on developing statistical models and machine learning algorithms for explanatory and predictive analysis of human activities in online systems. Of the large variety of human activities on the Web, we are particularly interested in three types of activities:

- **Decision making activities.** A large number of user activities in online systems can be seen as decision making behaviors, ranging from deciding which Webpage to browser, to choosing which URL to click from the search results. We are particularly interested in those scenarios where users are interacting with various online resources and deciding what to choose for consumption, for example, whenever they purchase a product, rent a movie, click an advertisement, indicate "like" to a celebrity, or participating an activity. The fundamental question is to understand why users make such a decision and to predict which will be the future decision of a particular user in a particular context.

- **Social interaction activities.** One of the most fascinating aspect of online systems is that it connects us together as a community, greatly facilitating interactions wherever we are and whatever we are doing. As a matter of fact, many emerging online systems include a social functionality in their systems, allowing users to connect to one another as friends, indicate trust/distrust, follow or share with one another, etc. Such social aspects have significant impact to individual behaviors. For example, because it enables users of various backgrounds to behave in one environment, the visibility of one user's behavior could potentially influence another user. This phenomena is widely known as "social contagion". Our goal is to understand social interactions and to capture the impact of social contagions in individual behavior of decision making, and in turn to improve the accuracy of behavior prediction in a social environment.

- **User generated contents.** We are also interested in understanding the contents created by users in online systems. Such user-generated contents include texts (e.g.,

status updates, tweets, blogs, comments, posts), images, videos, profiles, and even annotations (e.g., labels, tags, named entities, answers). Our goal is to identify what is the hidden intension behind a piece of content a particular user has generated, what this content suggests about the author or the difference among different authors, and how we can leverage the knowledge discovered to improve the system design to better serve the user.

1.1 Problems and Motivations

The main interests of our research are centered around these three types of online human activities, and accordingly focused on developing models and algorithms for modeling, understanding and predicting purposes. The main topics are three folds: (1) behavior prediction; (2) social contagion; and (3) content mining.

1.1.1 Behavior Prediction

The first topic of this thesis is concerned with *micro behavior prediction*, i.e., modeling the decision making behavior of each individual user as in a recommender system (e.g., Amazon, Netflix, Yahoo!, Facebook friend-finder, Google Adsense). We are particularly interested in predicting users' online choice behavior — when a user visits an online system (e.g., a website) and confronts with the resources (e.g., advertisement, product, movie) offered by the system, how shall he make choices (e.g., click, purchase, rent)? and how can we design more intelligent online systems to better serve users or to achieve certain strategic goals? Models and algorithms developed for this purpose are not only important for improving our understanding about human behavior but also of tremendous practical value to online industries to enhance the design of online systems, especially recommender systems.

Formally, consider a recommender system, where we have N users $u \in \mathcal{U} := \{1, 2, \ldots, N\}$ and M items $i \in \mathcal{I} := \{1, 2, \ldots, M\}$; when a user u visits the system, the system recommends a set of customized items $A = \{i_1, \ldots, i_l\}$ and u in turn chooses a (possibly empty) subset $R \subseteq A$ for consumption (e.g. buys some of the recommended products). Our goal in behavior prediction is to predict how an individual user u makes a decision R in a given context A, i.e., to quantify $p_u(R|A)$? And ultimately, how shall we design the recommender system — what recommendation A should be provided to a particular user u so as to satisfy the user's information need or to achieve certain strategic goals (e.g., maximum sales, revenue, profit)?

1.1.2 Social Contagion

We further examine decision making in a social environment (e.g., a social networking system) where online users actively make decisions while interacting with one another. Unlike recommender systems where users make their individual decisions independently, social contagion plays a significant role in online systems with social functionalities — users are connected in a community such that social interactions dramatically influence individual behavior of decision making; and behavior in turn influences social relations. Effective modeling of social contagion to capture the interplay between social interactions and individual behavior is not only of practical value to many internet services but also of general interest to the social science community. In particular, by capturing social contagion, we may improve social targeting to promote online services. For example, by providing better matched search results, News articles, games, advertisements, or products, one can improve

user satisfaction and also boost the revenue of a website (e.g. via product purchases, virtual transactions, advertisement clicks). Moreover, models for social contagion also have the potential to improve our understanding of social relations and our society, and in turn shed light to sociological principles in general.

Formally, consider a typical scenario in a social network where users routinely make decisions while actively interacting with one another. Particularly, we are given a set of users $u \in \mathcal{U} := \{1, 2, \ldots, N\}$ and a set of items (e.g., News articles, advertisement, retailing products, movies) $i \in \mathcal{I} := \{1, 2, \ldots, M\}$. The users are connected by a social network represented by the graph $\mathcal{G}(\mathcal{U}, \mathcal{C})$. Here $\mathcal{C} = \{c_{uv}\}$ denotes the set of edges: $c_{uv} \in \{1, \text{missing}\}$ that define the connection between a pair of users $(u, v) \in \mathcal{U}^2$. Users make decisions, for example, by clicking links, purchasing products, rating movies. Formally, a decision is a mapping $y : \mathcal{U} \times \mathcal{I} \to \mathcal{Y}$. That is: user u makes a decision regarding item i with a response $y_{ui} \in \mathcal{Y}$ (e.g. u rates movie i with a score of y_{ui}). Our goal is to characterize the interplay between individual behavior and social interactions, to identify hidden social aspects / mechanisms, and ultimately to improve social targeting, i.e., recommendation (e.g., services, resources, friends) in a social networking system.

1.1.3 Content Mining

The third topic of the thesis is on discovering knowledge from user-generated contents in online systems. We would like to infer knowledge *of* the user, *about* the user and *for* the user, i.e.: what are the intentions behind the contents? what do the contents suggest about the corresponding user? and what can we do to improve user experiences?

Of the various types of user-generated contents, we are particularly interested in texts, images, named entities and queries. Our interests are focused on the following three topics:

- *Mining hidden user aspects.* As users in an online system are extremely diverse in their background (e.g., age, nationality, race, income), one important research topic is how to discover knowledge about the users from the textual contents they have generated. Even more interestingly, is it possible to discover some subtle and hidden properties of social users such as culture, knowledgability, personality, life-style, mood, etc. from social texts? and how to identify and understand the differences in these aspects between different people?

- *Modeling social texts.* Social texts are more difficult to model than normal documents as they are mostly short, noisy and informal. How can we discover the user's intensions from such noisy texts and what are the more effective ways to analyze such texts? One of our research topics is on text representation, i.e., to develop effective models to capture more accurate information of a text and to facilitate our understanding of the text. We brought the pyramid representation and scale-space theory from image processing to textual domain. The proposed Language Pyramid (LaP) model [96] casts a document as a probabilistic distribution over the joint semantic-spatial space and efficiently encodes a document with a pyramid of matrices at a sequence of progressive discrete-scales. The scale-space theory for text [98] further extend this model toward continuous scales. We demonstrate how natural language documents can be understood, processed and analyzed at multiple resolutions, and how this scale-space representation can be used to facilitate a variety of text mining tasks. This work is, however, not included in the thesis, please refer to [96, 98] for more information.

- *Modeling complex-structure data.* One important phenomena we noticed is that many types of human activity data are structured in a much more complex form than what we usually see in traditional scenarios. For example, traditional machine learning models assume a nonambiguous scenario where both the input (i.e. predictors) and output (i.e. responses) of a system are *i.i.d.* points. However, the real-world is more like a web where input instances and/or output instances could be complexly connected/related to one another. Such data is particularly ubiquitous in online social media (e.g., social entities, texts, images, video, relations). This motivates us to investigate the generic complex-structure of such data and develop principled ways to handle it. Particularly, we examined ambiguous data structures in supervised learning. We established one of the first statistical models for ambiguous data [89, 88, 86]. The Dirichlet-Bernoulli Alignment (DBA) model [89] assumes a tree-structure for ambiguous data and encodes the tree with a Bayesian hierarchical model. The Exponential-Multinomial Mixture (EMM) model [88] further handles the challenge of label parsimoniousness with hybrid generative/ discriminative learning. This work has been demonstrated in text classification [89], entity extraction [89, 86], image annotation [88], and was successfully applied to `Bing` search engine for query disambiguation [86]. This part is not included in this thesis, interested readers may refer to [89, 88, 86] for more details.

1.2 Contributions and Organization

This dissertation focuses on the three topics of behavior prediction, social contagion and content mining. Accordingly, the thesis is organized as three parts, with each part dedicated to one topic.

We start with a brief introduction of related topics in Chapter 2. The reminder of the thesis is structured as follows:

- **Part I** *Behavior Prediction* (Chapter 3–4). The first part of the thesis is on behavior prediction. We present *collaborative competitive filtering* (CCF), a novel game-theoretic framework for recommendation. This part consists of two Chapters:
  - ◇ In Chapter 3, we present the CCF preference models, a set of models that learn user preference from the interactive choice process and provide satisfactory recommendations by preference based ranking [94].
  - ◇ In Chapter 4, we further present a game-theoretic formulation for recommendation, which enables us to optimize recommendation more strategically than what can be achieved by a simple preference based ranking. We therefore revisit the CCF model and extend it towards a game-theoretic framework [97].
- **Part II** *Social Contagion* (Chapter 5–6). The second part of the thesis is on social contagion, i.e., characterizing the interplay between social interactions and individual behavior of decision making. This part consists of two Chapters:
  - ◇ In Chapter 5, we examine the significance of the Homophily effect and present the joint *friendship-interest propagation* (FIP) model [93], a probabilistic model that leverages Homophily to address behavior prediction and link prediction simultaneously by coupling behavioral evidences with social interactions.
  - ◇ In Chapter 6, we examine signed social ties such as trust-distrust or friend-frenemy relationships, a much more complicated yet much stronger contagion

effect than Homophily. We establish the *behavior-relation interplay* (BIR) models [95], a series of models that characterize the social diffusion of behavior and the behavioral reflection into relationships. We also show how sociological principles such as the structure balance theory and the status theory can be encoded into the model and used to improve prediction accuracy.

- **Part III** *Content Mining* (Chapter 7). The third part of the thesis is on content mining. We establish a set of powerful tools to understand the contents user generated in online systems. This includes models for identifying user cognitive aspects [91, 18, 19], effective text representation [96, 97], and complex-structure (e.g., ambiguous) data [89, 88, 86], of which only the first one is included in this thesis. Particularly, in Chapter 7, we establish models for identifying user's hidden cognitive aspects, particularly the technicality/knowledgeability of a user from the texts he has generated. One direct motivation of the work is to address the language gap between layman people and experts. In particular, we seek to close the gap at the thematic level via *topic adaptation*, i.e., adjusting the topical structures for cross-domain documents according to a domain factor such as technicality and providing an effective topic-level bridge between lay and expert documents.

Finally in Chapter 8, we conclude and lay out a number of directions for future work.

CHAPTER II

PRELIMINARIES

In this chapter, we briefly review existing methodologies and related topics, which will provide a context of the work presented in this thesis. We start with an introduction of collaborative filtering, the most popular techniques for personalization and recommendation. We then provide a rough overview of some related topics on choice and game theories. These two will be used as foundations to the *collaborative competitive filtering* framework we present in Chapter 3 and 4. Next, we introduce some of the traditional methods for link prediction, which will be connected with the methodology we developed for social contagion as presented in Chapter 5 and 6. Finally, we briefly introduce topic modeling and *latent Dirichlet allocation* [10], which is a basic building block for the τLDA model we developed for mining user cognitive aspects, as will be presented in Chapter 7.

2.1 Collaborative filtering

Collaborative filtering (CF) is widely used for establishing personalized systems such as recommender systems. A typical setting for CF is to consider the rating prediction scenario, where we have N users $u \in \mathcal{U} := \{1, 2, \ldots, N\}$ and M items $i \in \mathcal{I} := \{1, 2, \ldots, M\}$; users express opinions (e.g., by assigning ratings) with respect to the items, and these largely-missing observations constitute a highly-incomplete matrix: $Y \in \mathcal{Y}^{N \times M}$, of which only a small number of entries are observed. Typically, a rating value is an integer value in the range 1 to 5, i.e.: $y_{ui} \in \mathcal{Y} = \{1, 2, 3, 4, 5\}$. Collaborative filtering predicts the values of unobserved entries by collaboratively uncovering the patterns underlying observed evidences and diffuse these patterns to the missing entries. In particular, collaborative filtering explores the notion of "collaboration effects" that similar users have similar preferences to similar items. By encoding collaboration, CF pools the sparse observations in such a way that for predicting $y(u, i)$ it also borrows observations from other (similar) users/items. Generally speaking, existing CF methods fall into either of the following two categories.

Neighborhood models. A popular approach to CF, commonly known as neighborhood models or memory based methods, is based on the principle of *locality of dependencies*, which assumes that the interaction between user u and item i can be restored solely upon the observations of neighboring users or items [72, 60]. Such neighborhood-based models therefore connect similar items to a particular user (item-oriented) or recommend a particular item to similar users (user-oriented). Basically, such approaches first define a similarity measure between items / users. Then, an unseen response between user u and item i is approximated based on the responses of neighboring users or items [72, 60], for example, by simply averaging the neighboring responses with similarities as weights. In particular, the user-oriented model uses:

$$\hat{y}_{ui} = \frac{\sum_{v \in \Omega_u} \omega_{uv} y_{vi}}{\sum_{v \in \Omega_u} \omega_{uv}},$$

where Ω_u is the set of neighboring (behaviorally similar) users to user u, ω_{uv} measures the degree of similarity, e.g. cosine score or Pearson correlation coefficient, between two users u

and v. Likewise, the item-based model uses:

$$\hat{y}_{ui} = \frac{\sum_{j \in \Omega_i} \omega_{ij} y_{uj}}{\sum_{j \in \Omega_i} \omega_{ij}},$$

where Ω_i is the neighbor set of item i, ω_{ij} is the similarity score between item i and j.

Latent factor models. The second class of methods for CF, known as matrix factorization methods or latent factor models, learn predictive latent factors as fingerprints to profile each user and each item and in turn estimate the missing dyadic responses based on the inner-products of latent factors. The basic idea is to associate latent factors, $p_u \in \mathbb{R}^k$ for each user u and $q_i \in \mathbb{R}^k$ for each item i, and assume a multiplicative model to approximate the dyadic response, i.e., $y_{ui} \approx q_u^\top q_i$ or $Y \approx PQ^\top$. This way the factors could explain observed ratings and in turn make prediction for the missing ones. An intuitive interpretation of this approach is to think of the latent aspects as item-clusters (e.g., "genres" of movie, "topics" of documents). For example, assume items could be grouped into a small number $k \ll \min\{M, N\}$ of "genres", we can think of the latent item profile q_i as the degrees of membership of the item i to each of these genres, and the user profile p_u as the *attitudes* of user u to each of the genres; accordingly, the rating y_{ui} is approximated by the inner-product $p_u^\top q_i$, i.e., the summation of the products of an item's association to a genre and the user's altitude toward that genre.

This model implicitly encodes the Aldous-Hoover theorem [36] for exchangeable matrices, i.e., it assumes that y_{ui} are independent from each other given p_u and q_i. Parameter estimation for the model reduces to a low-rank approximation of the matrix Y, which is given in a regularized form by:

$$\min_{P,Q} \frac{1}{2}||Y - PQ^\top||_F^2 + \frac{\lambda}{2}(||P||_F^2 + ||Q||_F^2).$$

The solution has a close connection with *singular value decomposition* (SVD) of Y:

$$Y = U\Sigma V^T,$$

where the matrices U and V are orthonormal and Σ is diagonal, i.e.:

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{\min\{N,M\}} \end{bmatrix}.$$

The values σ_1, σ_2, ..., $\sigma_{\min\{N,M\}}$ are the singular values of the matrix Y. Without loss of generality, we assume that the singular values are arranged in descending order, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{N,M\}}$. The two factors P and Q can be obtained from the rank-k approximation of Y, which can be obtained with a partial SVD using the singular vectors corresponding to the k largest singular values of Y:

$$\begin{aligned} PQ^\top = \hat{Y} &= \hat{U}\hat{\Sigma}\hat{V}^T \\ &= \begin{bmatrix} u_1 & \dots & u_k \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_k^T \end{bmatrix}. \end{aligned}$$

For the above reason, the approach is also commonly referred to in the CF literature as SVD. The rank-k approximation $\hat{Y}$ has the minimum distance from Y in terms of the spectral norm and the Frobenius norm. Note that although Y is typically sparse, $\hat{Y}$ is generally not. Thus, $\hat{Y}$ can be viewed as a diffused version of Y.

Latent factor models have gained tremendous successes in recommendation systems and have even become the current state-of-the-art for CF [41, 1]. A known drawback for such models is that, because it is learned only upon past interactions, the generalization performance is usually poor for completely new entities, i.e. unseen users or items, for which the observations are missing at the training stage. This scenario is well-known as the "cold-start problem" in recommendation systems. The recently proposed *regression based latent factor model* (RLFM) [1] addresses this problem by incorporating entity features into latent factor learning. The key idea is to use observable features to explain the learned latent variables (e.g. by regression or factorization). Suppose for each user and each item, there are observable features, x_u for user u (e.g. user's demographic information, self-crafted registration profiles) and x_i for item i (e.g. content of a document, description of a product), RLFM [1] assumes a regression model between the latent profile p_u and observed feature x_u, and similarly between q_i and x_i.

It is natural to combine the neighborhood models and latent factor models. A recent example is discussed [40], where the basic idea is to apply the *locality of dependencies* directly to the latent factors, for example:

$$\hat{p}_u = \frac{\sum_{v \in \Omega_u} \omega_{uv} p_v}{\sum_{v \in \Omega_u} \omega_{uv}} \qquad\qquad \hat{q}_i = \frac{\sum_{j \in \Omega_i} \omega_{ij} q_j}{\sum_{j \in \Omega_i} \omega_{ij}}$$

This model which is quite similar to [40] was deployed on the Netflix data yielding significantly better performances over both pure-neighborhood and pure latent factor models.

2.2 Choice and game theory

In this section, we provide a brief introduction to some related concepts and topics in rational choice and game theories.

Rational choice. Rational choice theory is a formal theoretic framework for understanding human economic behavior (i.e., decision making) [59, 30]. The basic idea is that people make decisions by comparing the costs and benefits of each decision option and picking the one that maximizes their benefit and minimizes their costs.

Consider a decision scenario where an agent u is confronted with a set of exhaustive and exclusive alternatives $\mathcal{I} = \{1, \ldots, M\}$, let U_i be the utility, i.e., a measure of satisfaction received by u from consuming item $i \in \mathcal{I}$. U_i defines a preference relationship among the available choices. Rational choice theory makes two fundamental axiomatic assumptions about individuals' preferences for choices:

COMPLETENESS: A preference relation $\succcurlyeq$ is said to be complete if all the choice options can be ranked in a consistent order of preference based on $\succcurlyeq$. In other words, for any two options $i \in \mathcal{I}$ and $i' \in \mathcal{I}$, either $i \succcurlyeq i'$ or $i' \succcurlyeq i$ unless they are equivalent $i \sim i'$ (indifferent).

TRANSITIVITY: A preference relationship $\succcurlyeq$ is transitive if for three options i_1, i_2 and i_3, $i_1 \succcurlyeq i_2$ together with $i_2 \succcurlyeq i_3$ leads to $i_1 \succcurlyeq i_3$.

These assumptions together enable an individual to rank choice options consistently in terms of his preferences. Given a set of mutually exclusive decision options, the rational choice theory states that the agent makes a rational decision by picking the one that provides

the greatest benefit at the lowest cost. In particular, a rational decision maker considers both opportunity cost and revenue, and decides his choice by comparing the economic profit of each option.

REVENUE: The revenue is a measure of the gross benefit received from consuming a good. We denote the revenue from item i as r_i.

OPPORTUNITY COST: Opportunity cost is the cost of taking one action that excludes the agent to try other alternatives. It is measured in terms of the value of the next best alternative that is not chosen. The opportunity cost can be quantified by

$$c_i = \max_{j \in \mathcal{I}, j \neq i} r_i.$$

ECONOMIC PROFIT: Economic profit is the net benefit an agent u received from choosing and consuming an option i. It arises only when revenue exceeds the opportunity cost. Specifically, we have the profit $\pi_i = r_i - c_i$.

Choice under risk. In the above, we assume the agent has full or perfect information about what will occur from his choice decision. In many realistic scenarios, however, different outcomes occur with probabilities such that the decision maker has to choose between risky or uncertain prospects, similar to what happens in gambling.

Let us consider the scenario of decision making in an uncertain environment, where a decision maker has preferences over decision options where the outcomes realize with objective probabilities that are known to the decision maker. A choice option is usually called a lottery in such a case.

SIMPLE LOTTERY: Given a finite set of deterministic outcomes $\{z_1, \ldots, z_n\}$, a simple lottery $\vec{p}$ is a n-tuple $\vec{p} = (p_1, \ldots, p_n)$ with $p_i \geqslant 0$ for all $i \in \{1, \ldots, n\}$ and $\sum_{i=1}^{n} p_i = 1$, where p_i is interpreted as the probability that outcome z_i occurs. Denote δ_i the degenerated lottery that realize z_i with probability 1, and $\mathcal{P}$ the set of all simple lotteries over $\{z_1, \ldots, z_n\}$, we have: $\mathcal{P} = \{\vec{p} \in \mathbb{R}_+^n | \sum_{j=1}^{n} p_j = 1\}$, i.e., the $(n-1)$-dimensional simplex.

COMPOUND LOTTERY: A lottery can also be a compound of multiple simple choices. Given K simple lotteries, $\{\vec{p}_1, \ldots, \vec{p}_K\}$, a compound lottery $\vec{p}_\alpha = \sum_{k=1}^{K} \alpha_k \vec{p}_k$, where $\alpha_k \geqslant 0$ and $\sum_{k=1}^{K} \alpha_k = 1$. It can be shown that a compound lottery can be reduced to a simple lottery $\vec{p}$ where $p_i = \sum_{k=1}^{K} \alpha_k p_{ki}$.

In the presence of risky or uncertain outcomes, a decision maker could use the expected utility hypothesis to decide his choice. The von Neumann-Morgenstern utility theorem [64] provides some necessary and sufficient axioms under which this expected utility criterion should work. As a result, the rationality of such a decision is called von Neumann-Morgenstern rationality [64].

Given a utility function $U : \mathcal{P} \to \mathbb{R}$, which induces a preference relation $\succcurlyeq$ over the lotteries in the sense that $\vec{p}_1 \succcurlyeq \vec{p}_2$ if and only if $U(\vec{p}_1) \geqslant U(\vec{p}_2)$. In addition to the completeness and transitivity axioms we have already described in the deterministic case, we need two more assumptions for the scenario of decision making under uncertainty:

CONTINUITY: Let $\vec{p}_1$, $\vec{p}_2$ and $\vec{p}_3$ be three lotteries with $\vec{p}_1 \succcurlyeq \vec{p}_2 \succcurlyeq \vec{p}_3$, then there exists a constant $0 \leqslant \alpha \leqslant 1$ such that $\vec{p}_2 \sim \alpha\vec{p}_1 + (1-\alpha)\vec{p}_3$. In essence, this assumption states that the preference relation over lotteries is continuous.

INDEPENDENCE: Let $\vec{p}_1$ and $\vec{p}_2$ be two lotteries with $\vec{p}_1 \succcurlyeq \vec{p}_2$, and $\vec{p}_3$ be another lottery, then for any constant $0 < \alpha \leqslant 1$, the preference over $\vec{p}_1$ and $\vec{p}_2$ will not change if compounded with $\vec{p}_3$ at the same proportions, that is $[\alpha\vec{p}_1 + (1-\alpha)\vec{p}_3] \succcurlyeq [\alpha\vec{p}_2 + (1-\alpha)\vec{p}_3]$.

If all the four axioms are satisfied, the von Neumann-Morgenstern theorem [64] states that the rational decision could be achieved by maximizing the expected utility.

[Expected Utility Theorem]: *Suppose $\succcurlyeq$ is a preference that satisfies the 4 axioms, for any two lotteries $\vec{p}_1$ and $\vec{p}_2$, we have: $\vec{p}_1 \succcurlyeq \vec{p}_2$ if and only if $\mathbb{E}_{\vec{p}_1}(U) \geqslant \mathbb{E}_{\vec{p}_2}(U)$.*

Interactive choice and game theory. We now move our discussion from the topic of individual's decision making to the more complicated case involving multiple decision makers, where the decisions of different players are interdependent and examining the decision in such an interactive scenario need tools from game theory.

Game theory is an area in applied mathematics that studies and analyzes the decision strategies among a number of interacting rational decision makers (players). It has been widely used to study (e.g., describe, predict, explain, model) a large variety of human behaviors in many disciplines in social sciences, e.g., psychology, economics, philosophy and political science. In terms of different aspects, games studied in game theory can be classified into different types, for example, cooperative vs non-cooperative games, zero-sum vs non-zero-sum games, symmetric vs asymmetric games, simultaneous vs sequential games. In Chapter 4, we show that the user-system interactions on recommender system can be formulated as a collection of non-cooperative sequential games. We do not intend to provide a thorough introduction to game theory, but rather, we would like to give a quite brief description of the related concepts and the formulation in non-cooperative games. For more detailed descriptions, please refer to any textbook on game theory, for example, the excellent yet concise book by Leyton-Brown and Shoham [49].

In a strategic form, a game consists of two or more players; each player has a set of strategies; for each combination of strategies (i.e., outcome), there is a numerical payoff for each player that reflects the benefit the player obtains from an outcome. We assume each players is a rational decision maker who seeks to maximize his payoff. A game is said to be a cooperative game if groups of players form coalitions and the competitions are among the coalitions rather than individuals. Otherwise, the game is non-cooperative, i.e., players make decision individually. Formally, a non-cooperative game $\mathcal{G}$ can be formulated as a triple $(P, \mathcal{Z}, U)$, where $P = \{p_1, \ldots, p_n\}$ is the set of players where $n = |P|$, $\mathcal{Z}_i$ is the strategy space for player p_i, i.e., the set of all actions (decision options) available to p_i, the joint strategy space $\mathcal{Z} = \mathcal{Z}_1 \times \ldots \times \mathcal{Z}_n$. The concatenation of the selected strategies by all the game players constitutes a strategy profile or outcome $z \in \mathcal{Z}$. The payoff function for player p_i is denoted by $U_i(z)$ which measures the utility of an outcome z to player p_i, the joint payoff function $U(z) = (U_1(z), \ldots, U_n(z))$ for any $z \in \mathcal{Z}$. The best action strategy of p_i is defined by the strategy $z_i \in \mathcal{Z}_i$ that maximizes player p_i's utility U_i given other players' strategies. As a result, to find the best strategy for a particular player, one needs to predict other players' strategies in advance and maximizes the targeted player's payoff thereafter. In this spirit, we propose the *collaborative competitive filtering* framework in Chapter 4 that predicts users' behaviors in advance and in turn maximizes system's payoff. The games we examined is sequential in the sense that the players do not move simultaneously. As a matter of fact, since users choose from the recommendations by the system, at the time a user moves, he knows everything about the system's strategy, therefore, the interaction at recommender systems is actually a sequential game with perfect information.

2.3 Link prediction

One of the core properties of a social network site is the social graph, represented by $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, where the vertex set $\mathcal{U}$ consists of all the registered users, and the edge set $\mathcal{E}$ consists all the relationship / connections among users. One key task of fundamental importance to a social network system is link prediction or friendship suggestion, i.e., recommends users

to other users in the hope of acquainting people who were previously not connected in the network (or even unfamiliar with each other). Link prediction crucially influences the traffic, user base and revenue of a social network and is hence recognized as one of the key tasks in social network analysis.

Node proximity and random walk. Consider a graph $\mathcal{G}$ with N vertices $\mathcal{U} = \{1, \ldots, N\}$. A convenient way to represent a graph is to use the adjacency matrix $C_{N\times N}$, where $c_{uv} = 1$ if $(u, v) \in \mathcal{E}$ or 0 otherwise. If $\mathcal{G}$ is an undirected graph, C is a symmetric matrix. The basic idea underlying many link prediction algorithms is to assess the similarity / proximity between any pair of vertices (u, v) based on the observed edges of the input graph. For example, a simple measure could be the number of common neighbors or its normalized version, i.e., the Jaccard similarity. The problem with these simple measures is that they can only capture local second-order information. A more reliable measure that is able to capture higher-order information about the network topology is the length of shortest path, which is however computationally expensive. Many existing approaches therefore employ random walk [50, 70] and assess the similarity based on diffusion probability or hitting time.

A random walk on the graph $\mathcal{G}$ is a reversible Markov chain on the vertices $\mathcal{U}$. The transition probability from the vertex u to vertex v is denoted $p_{uv} = p(v|u)$. Vertices are considered close whenever the hitting time is small or whenever the diffusion probability is large. Let c_{uv} be the connection weight between vertices u and v, and $d_u = \sum_v c_{uv}$ the degree of vertex u, one reliable way to obtain p_{uv} is to consider random walk with restart:

$$p_{uv} = \lambda \frac{c_{uv}}{d_u} + (1-\lambda)\delta_{uv} \quad \text{or} \quad p_u = \lambda \tilde{C} p_u + (1-\lambda) e_u$$

where $p_u = [p_{u1}, \ldots, p_{uN}]^\top$, e_u is the u-th column of an $N \times N$ identity matrix, $\tilde{C} = (\tilde{c}_{uv})$ where $\tilde{c}_{uv} = c_{uv}/d_u$. Basically, this formulation assumes that with probability λ the traveler will go from vertex u to one of the neighboring vertices with probability proportional to their connection weight, and with probability $1 - \lambda$ it will stay at the current vertex.

Accordingly, the proximity measure p_{uv} can be solved directly from the formulation as:

$$p_u = (1-\lambda)(I - \lambda\tilde{C})^{-1} e_u.$$

A simple way to obtain p_u is to iteratively carry out the update above until convergence. Note that the inverse $(I - \lambda\tilde{C})^{-1}$ only need to be computed once.

Latent factor network model. Recent advances in link prediction developed factorization models for networks based on the very same idea as in collaborative filtering. The basic idea is again low-rank matrix factorization — given the adjacency matrix C, we factorize C in terms of low-rank factors, i.e., $\hat{C} = PQ^\top$ for directed graph or $\hat{C} = PP^\top$ for undirected graph. These factors are learned by fitting on the current topology of the network, for example, by minimizing the Frobenius norm of the residue matrix $(\hat{C} - C)$. Nonetheless, since the connections are usually binary, i.e., $c_{uv} = 1$ or missing, a logistic loss function is more reasonable than ℓ_2. Therefore, we usually use (for undirected graph):

$$\min \sum_{(u,v)\in\mathcal{E}} \log\Big[1 + \exp(-p_u^\top p_v)\Big] + \lambda \sum_{u\in\mathcal{U}} ||p_u||^2$$

This way the factors fully characterize the network topology and in turn predicts how likely the emergence of a edge is, e.g., through logistic regression mapping of the inner-product $p_u^\top p_v$.

A slightly different approach is to use a three-factor low-rank model, i.e., $\hat{C}_{N\times N} = P_{N\times k}B_{k\times k}P^{\top}_{N\times k}$. This model can be though of as clustering vertices into k clusters (or communities), where p_u can be viewed as the memberships of vertex u to each of these communities, and H as the base connections, which is the connection matrix between communities instead of individual vertices. Here, we briefly review one representative of such generative models for undirected graph, i.e., the Mixed Membership Stochastic Blockmodels (MMSB) [2]. This model casts each vertex factor as a multinomial distribution and poses the conjugate (Dirichlet) prior to obtain a full Bayesian model. MMSB assumes that each vertex can be associated with multiple communities but in a particularly relationship (i.e., a particular edge), it is only associated with one of these communities. Given a network graph $\mathcal{G}$ with vertex set $\mathcal{U} = \{1, \ldots, N\}$ and adjacency matrix $C_{N\times N}$, let $\theta_u \in \mathbb{R}^k$ denote the latent factor for vertex u, MMSB assumes the following generative process for $\mathcal{G}$:

- For each vertex u:
  - draw vertex factor (i.e., mixed membership vector): $\theta_u \sim \text{Dir}(\alpha)$
- For each pair of vertices, (u, v):
  - draw membership indicator for the initiator vertex $z_{u\to v} \sim \text{Multinomial}(\theta_u)$;
  - draw membership indicator for the receiver vertex $z_{v\to u} \sim \text{Multinomial}(\theta_v)$;
  - draw connection weight $c_{uv} \sim \text{Bernoulli}(z_{u\to v}Bz_{v\to u})$

The model can be learned based on the current topology of the network and inference about new connections can be done simply through logistic regression mapping of $\theta_u^{\top}B\theta_v$.

2.4 Topic modeling

The third part of the thesis is on mining user-generated contents, especially texts. Probabilistic topic models have emerged as an important set of tools for text mining (e.g., automatic organizing, browsing, managing). In this section, we briefly review *latent Dirichlet allocation* (LDA) [10], which will be used as a building block in Chapter 7 of the thesis for mining user-generated texts.

Latent Dirichlet allocation. The basic idea underlying probabilistic topic models is dimensionality reduction in a probabilistically comprehensible way. It is essentially the very same idea underlying the latent factor models in collaborative filtering and many other techniques. Basically, given a document-word matrix $Y_{M\times N}$, where M is the total number of documents and N the total number of words (i.e., the size of the vocabulary), and each row vector of Y is the bag-of-word (TF vector) representation of a document, the idea is to decompose Y in terms of two low rank matrix $P_{M\times k}$ and $Q_{N\times k}$, where $k \ll M$ or N. If we interpret the latent aspects of Q as k term-clusters or *topics* and P as the association degrees of each document to these topics, such factorization leads to a semantically coherent low-rank embedding of documents and words. As a matter of fact, this is the very idea that underly both the matrix factorization model in collaborative filtering [41] and the *latent semantic index* (LSI) model for topic modeling [20]. The *probabilistic latent semantic index* (PLSI) model [33] provides a probabilistic extension to the LSI model by casting each column vector of Q and each row vector of P as multinomial distributions. This probabilistic view is favorable as it greatly facilitates the semantical interpretation of the latent topics. The latent Dirichlet allocation (LDA) model [10] further extends PLSI

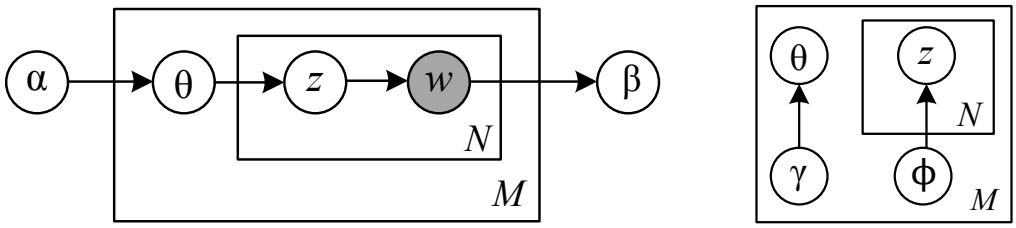


Figure 1: Latent Dirichlet allocation (LDA) and its variational model.

towards a Bayesian model by imposing conjugate priors, i.e., the Dirichlet distribution, to each multinomial. The Dirichlet priors are useful as they help model selection and is also useful to avoid overfitting.

Fig. 1 shows the diagram of the graphical model representation of LDA. In the diagram, each random variable is represented by a circle. A variable that is *observed* (its outcome is known) is shaded. An arrow is drawn from one random variable to another if the the outcome of the second variable depends on the value of the first variable. A rectangular plate is drawn around a variable to show that it is repeated multiple times. Given a corpus $\mathcal{D}$ consisting of M documents, with each document d_m being a finite sequence of words $w_1 w_2 \dots w_{N_m}$, assume the vocabulary size is W, and the number of topics is k; and to comply with the topic modeling convention, we use θ instead of P as document profile (i.e., topic distributions for documents) and β instead of Q as word profile (i.e., topic bases); the LDA model assume the following process for generating a document:

- Choose the term probabilities for each topic. The distribution of terms for each topic i is represented as a multinomial distribution β_i, which is drawn from a symmetric Dirichlet distribution with parameter η.

$$\beta_i \sim \mathrm{Dir}(\eta)\,; \qquad p(\beta_i|\eta) = \frac{\Gamma(W\eta)}{[\Gamma(\eta)]^W} \prod_{v=1}^{W} \beta_{iv}^{\eta-1}\,.$$

- Choose the topics of the document. The topic distribution for document d is represented as a multinomial distribution θ_d, which is drawn from a Dirichlet distribution with parameters α.

$$\theta_d \sim \mathrm{Dir}(\alpha)\,; \qquad p(\theta_d|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_{di}^{\alpha_i - 1}\,.$$

- For each of the token w_n:
  - Choose the topic assignment for each token. The topic z_{dn} for each token index n is chosen from a discrete distribution based on the document-specific topic distribution θ_d.

$$z_{dn} \sim \mathrm{Multinomial}(\theta_d)\,; \qquad p(z_{dn} = i|\theta_d) = \theta_{di}\,.$$

  - Choose each token. Each token w is chosen from the multinomial distribution associated with the selected topic.

$$w_{dn} \sim \text{Multinomial}(\beta_{z_{dn}})\,; \qquad p(w_{dn} = v | z_{dn} = i, \beta_i) = \beta_{iv}\,.$$

LDA provides the mechanism for finding patterns of term co-occurrence and using those patterns to identify coherent topics. As a result of the LDA generative process, all terms that co-occur frequently will be grouped together and appear with high probabilities in a topic.

Training an LDA model involves finding the optimal set of parameters, under which the probability of generating the training documents is maximized. The likelihood of a given corpus $\mathcal{D}$ under LDA model is given by:

$$\mathcal{L} = \prod_{d=1}^{M} \prod_{n=1}^{N_m} p(w_{dn_m} | z_{dn_m}, \beta) p(z_{dn_m} | \theta_d) p(\theta_d | \alpha) p(\beta | \eta).$$

Unfortunately, the direct optimization of the likelihood is problematic because the topic distribution θ_d and per-word assignments z_{dn} are not directly observed and need to be marginalized. Even inference for a single document is intractable. Standard LDA training algorithms including *collapsed Gibbs sampling* algorithm and *variational Bayes* approach. Here, we briefly review the latter, which approximates the model with a series of simpler models but neglect the troublesome dependencies.

Variational inference. Variational approach approximates the true posterior distribution of the latent variables by a fully-factorized distribution—this proxy is usually referred to as the variational model (cf Fig. 1), which assumes all the latent variables are independent of each other. Particularly, we assume the following variational model for the posterior of z and θ:

$$q(\theta, Z | \gamma, \phi) = q(\theta | \gamma) \prod_{n=1}^{N} q(z_n | \phi_n) = \text{Dir}(\theta | \gamma) \prod_{n=1}^{N} \text{Multinomial}(z_n | \phi_n).$$

Essentially, this variational distribution is a simplification of the original LDA graphical model by removing the edges between the vertices θ and Z (Fig. 1). The optimal approximation is achieved by optimizing the distance (for example, the KL divergence) between the true model and the variational model:

$$\min_{\gamma, \phi} \quad KL[q(\theta, Z | \gamma, \phi) \;||\; p(\theta, Z | \alpha, \beta)]\,.$$

It can be shown that the above KL-divergence is the discrepancy between the true log-likelihood and its variational lower-bound that is used in the variational EM algorithm (described later in this section) for estimating the LDA hyperparameters α and β.

The optimization has no close-form solution but can be implemented through iterative updates,

$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni}, \quad \phi_{ni} \propto \beta_{iw_n} \exp[\Psi(\gamma_i)]\,,$$

where $\Psi(\cdot)$ is the bi-gamma function.

Variational EM for LDA training We learn a LDA topic model by maximizing the likelihood of the corpus.

$$\max_{\alpha, \beta} \; \sum_{d=1}^{M} \log p(d | \alpha, \beta) = \max_{\alpha, \beta} \; \sum_{d=1}^{M} \log \int_{\theta_d, Z_d} p(d, \theta_d, Z_d | \alpha, \beta) d\theta_d dZ_d\,.$$

The variational approximation provides a tractable lower bound for this likelihood function:

$$\begin{aligned}\mathcal{L}(\gamma, \phi) &= \log p(d|\alpha, \beta) - KL(q(Z, \theta|\gamma, \phi)||p(Z, \theta|\alpha, \beta)) \\ &\leq \log p(d|\alpha, \beta)\,,\end{aligned}$$

where $\mathcal{L}(\gamma, \phi) = \mathbb{E}_q[\log p(w_d, \theta, Z) - \log q]$ is the variational lower bound for the log-likelihood. The maximum likelihood estimation therefore involves a two-layer optimization,

$$\max_{\alpha,\beta} \sum_{d=1}^{M} \max_{\gamma_d, \phi_d} \mathcal{L}(\gamma_d, \phi_d)\,.$$

The inner-loop (the optimization with respect to γ and ϕ, referred to as the Variational E-step) goes through the whole corpus and performs variational approximation for each of the documents, which ends up with a tight lower bound for the log-likelihood. Then the M-step updates the model parameters (α and β) by optimizing this lower-bound approximation of the log-likelihood. The E- and M-steps are alternated in an outer loop until convergence.

CHAPTER III

COLLABORATIVE COMPETITIVE FILTERING I

The first part of the thesis is concerned with *micro behavior prediction*, i.e., modeling the decision making behavior of each individual user as in a recommender system (e.g., Amazon, Netflix, Yahoo!, Facebook friend-finder, Google Adsense). This includes this and the next two chapters. We are particularly interested in predicting users' online choice behavior — when a user visits a online social media site and confronts with the resources (e.g., advertisement, product, movie) offered by the system, how shall he make choices (e.g., click, purchase, rent)? Models and algorithms developed for this purpose are of tremendous practical value to personalized recommendation, advertisement targeting, web search, etc.

We propose *Collaborative Competitive Filtering* (CCF) [90, 94, 97], a game-theoretic framework for recommendation. The framework consists of two parts. This section describe the first part, which is composed of a series of preference models based on the interactive choice process in recommender systems. In next chapter, we formally describe the game-theoretic formulation and presented the second part of the framework, i.e., optimizing recommendations in respect of certain strategic goals.

The proposed CCF preference models employ a multiplicative latent factor model to characterize the dyadic utility function. But unlike existing collaborative filtering (CF) approaches, CCF models the user behavior of choices by encoding a local competition effect. In this way, CCF allows us to leverage dyadic data that was previously lumped together with missing data in existing CF models. We present two formulations and an efficient large scale optimization algorithm. Experiments on three real-world recommendation data sets demonstrate that CCF significantly outperforms standard CF approaches in both offline and online evaluations.

3.1 Introduction

Recommender systems have become a core component for today's personalized online businesses. With the abilities of connecting various items (e.g., retailing products, movies, News articles, advertisements, experts) to potentially interested users, recommender systems enable online firms (e.g. Amazon, Netflix, Yahoo!) to expand the marketing efforts from historically a few best-sellings toward a large variety of long-tail (niche) products [12, 79, 22]. Such abilities are endowed by a personalization algorithm for identifying the preference of each individual user, which is at the heart of a recommender system.

Predicting user preference is challenging. Usually, the user and item spaces are very large yet the observations are extremely sparse. Learning from such rare, noisy and largely missing evidences has a high risk of overfitting. Indeed, this *data sparseness* issue has been widely recognized as a key challenge for constructing effective recommender systems.

A straightforward way for personalization would be to learn each user's preference based on the prior interactions between the user and the system. Typically, such interaction is an "opportunity give-and-take" process (*c.f.* Table 1), where at each interaction:

1) a user u inquires the system (e.g. visits a movie recommendation web site);

Table 1: An example of user-system interactions and the degraded dyadic matrix.

User	Offer set	Choice
u_1	$[i_1, i_2, i_3, i_5]$	i_2
u_2	$[i_2, i_3, i_4, i_5]$	i_2
u_3	$[i_1, i_3, i_5, i_6]$	i_5
u_4	$[i_2, i_3, i_4, i_6]$	i_3
u_5	$[i_1, i_3, i_4, i_5]$	i_4
u_6	$[i_1, i_4, i_5, i_6]$	i_6

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	·	1	·		·	
u_2		1	·	·	·	
u_3	·		·		1	·
u_4		·	1	·		·
u_5	·		·	1	·	
u_6	·			·	·	1

2) the system offers a set of (personalized) opportunities (i.e. items) $\mathcal{O} = \{i_1, \ldots, i_l\}$ (e.g. recommends a list of movies of potential interest to the user);
3) the user chooses one item $i^* \in \mathcal{O}$ (or more) from these offers and takes actions accordingly (e.g. click a link, rent a movie, view a News article, purchase a product).

Somewhat surprisingly, this interaction process has not been fully-exploited for learning recommenders. Instead, research on recommender systems has focused almost exclusively on recovering user preference by completing the matrix of user actions (u, i^*) while the actual contexts in which user decisions are made are totally disregarded. In particular, Collaborative Filtering (CF) approaches only captures the action dyads (u, i^*) while the contextual dyads (i.e. $\{(u, i)\}$ for all $i \in \mathcal{O}$ and $i \neq i^*$) are typically treated as missing data. For example, the rating-oriented models aim to approximating the ratings that users assigned to items [72, 60, 1, 41]; the recently emerged ranking-oriented algorithms [84, 51] attempt to recover the ordinal ranking information derived from the ratings. Although this matrix-completion formulation of the recommendation problem has led to numerous algorithms which excel at a number of data sets, including the prize-winning work of [41], we argue here that the formulation is inherently flawed — a preference for *Die Hard* given a generic set of movies only tells us that the user appreciates action movies; however, a preference for *Die Hard* over *Terminator* or *Rocky* suggests that the user might favor Bruce Willis over other action heroes. In other words, the context of user choice is vital when estimating user preferences.

When it comes to modeling of user-recommender interactions, an important question arises: what is the fundamental mechanism underlying the user choice behaviors? As reflected by its name, collaborative filtering is based on the notion of "collaboration effects" that *similar items get similar responses from similar users*. This assumption is essential because by encoding the "collaboration" among users or among items or both, CF greatly alleviates the issue of data sparseness and in turn makes more reliable predictions based on the somewhat *pooled* evidences.

It has long been recognized in psychology and economics that, besides the effect of collaboration [14, 61], another mechanism governs users' behavior — *competition* [53, 59, 4]. In particular, items turn to compete with each other for the attention of users; therefore, axiomatically, user u will pick the *best* item i^* (i.e. the one with highest utility) when confronted by the set of alternatives $\mathcal{O}$. For example, consider a user with a penchant for action movies by Arnold Schwarzenegger. Given the choice between *Sleepless in Seattle* and *Die Hard* he will likely choose the latter. However, when afforded the choice between the oeuvres of Schwarzenegger, Diesel or Willis, he's clearly more likely to choose Schwar-zenegger over the works of Willis. To capture user's preference more accurately, it is therefore *essential* for a recommender model to take into account such local competition effect. Unfortunately, this effect is absent in a large number of collaborative filtering approaches.

In this chaper, we present Competitive Collaborative Filtering (CCF) for learning recommender models by modeling users' choice behavior in their interactions with the recommender system. Similar to matrix factorization approaches for CF, we employ a multiplicative latent factor model to characterize the dyadic utility function (i.e. the utility of an item to a user). In this way, CCF encodes the collaboration effect among users/items just as CF does. But instead of learning only the action dyads (i.e. (u, i^*) or the "1" entries in Table 1), CCF bases the learning of the factorization on the whole user-recommender interaction sessions. It therefore leverages not only the action dyads (u, i^*) but also the dyads in the context without user actions (i.e. (u, i) for all $i \in \mathcal{O}$ and $i \neq i^*$, or the dot entries in Table 1), which were treated as potentially missing data in CF approaches.

To leverage the entire interaction session for latent factor learning, we devise probabilistic models or optimization objectives to encode the local competition effect underlying the user choice process. We present two formulations with different flavors. The first formulation is derived from the *multinomial logit model* that has been widely used for modeling user choice behavior (e.g. choice of brands) in psychology [53], economics [56, 59] and marketing science [27]. The second formulation relates closely to the ordinal regression models in content filtering [32] (e.g. web search ranking). Essentially, both formulations attempt to encodes "local optimality of user choices" to encourage that every opportunity i^* taken by a user u be locally the best in the context of the opportunities $\mathcal{O}$ offered to her. From a machine learning viewpoint, CCF is a hybrid of *local* and *global* learning, where a global matrix factorization model is learned by optimizing a local context-aware loss function. We discuss the implementation of CCF, establish efficient learning algorithms and deliver a package that allows distributed optimization on streaming data.

Experiments were conducted on three real-world recommendation data sets. First, on two dyadic data sets, we show that CCF improves over standard CF models by up to 50+% in terms of offline top-k ranking. Furthermore, on a commercial recommender system, we show that CCF significantly outperform CF models in both offline and online evaluations. In particular, CCF achieves up to 7% improvement in offline top-k ranking and up to 13% in terms of online click rate prediction.

3.2 Problem definition

Consider the user-system interaction in a recommender system: we have users $u \in \mathcal{U} := \{1, 2, \ldots, U\}$ and items $i \in \mathcal{I} = \{1, 2, \ldots, I\}$; when a user u visits the site, the system recommends a set of items $\mathcal{O} = \{i_1, \ldots, i_l\}$ and u in turn chooses a (possibly empty) subset $\mathcal{D} \subseteq \mathcal{O}$ from $\mathcal{O}$ and takes actions accordingly (e.g. buys some of the recommended products). For ease of explanation, let us temporarily assume $\mathcal{D} = \{i^*\}$, i.e. $\mathcal{D}$ is not empty and contains exactly one item i^*. More general scenarios shall be discussed later.

To build the recommender system, we record a collection of historical interactions in the form of $\{(u_t, \mathcal{O}_t, \mathcal{D}_t)\}$, where t is the index of a particular interaction session. Our goal is to generate recommendations $\mathcal{O}_{\tilde{t}}$ for an incoming visit $\tilde{t}$ of user $u_{\tilde{t}}$ such that the user's satisfaction is maximized. Hereafter, we refer to $\mathcal{U}$ as *user space*, $\mathcal{I}$ as *item space*, $\mathcal{O}_t$ as *offer set* or *context*, $\mathcal{D}_t$ as *decision set*, and i^* as a *decision*.

A key component of a recommender system is a model $r(u, i)$ that characterizes the utility of an item $i \in \mathcal{I}$ to a user $u \in \mathcal{U}$, upon which recommendations for a new inquiry from user u could be done by simply ranking items based on $r(u, i)$ and recommending the top-ranked ones. Collaborative filtering is by far the most well-known method for modeling such dyadic responses.

3.3 Collaborative filtering

In collaborative filtering we are given observations of dyadic responses $\{(u, i, y_{ui})\}$ with each y_{ui} being an observed response (e.g. user's rating to an item, or indication of whether user u took an action on item i). The whole mapping:

$$(u, i) \to y_{ui} \text{ where } u \in \mathcal{U}, i \in \mathcal{I}$$

constitutes a large matrix $Y \in \mathcal{Y}^{|\mathcal{U}|\times|\mathcal{I}|}$. While we might have millions of users and items, only a tiny proportion (considerably less than 1% in realistic datasets) of entries are observable[1].

Collaborative filtering explores the notion of "collaboration effects", i.e., similar users have similar preferences to similar items. By encoding collaboration, CF pools the sparse observations in such a way that for predicting $r(u, i)$ it also borrows observations from other (similar) users/items. Generally speaking, existing CF methods fall into either of the following two categories.

3.3.1 Neighborhood models

A popular class of approaches to CF is based on propagating the observations of responses among items or users that are considered as neighbors. The model first defines a similarity measure between items / users. Then, an unseen response between user u and item i is approximated based on the responses of neighboring users or items [72, 60], for example, by simply averaging the neighboring responses with similarities as weights.

3.3.2 Latent factor models

This class of methods learn predictive latent factors to estimate the missing dyadic responses. The basic idea is to associate latent factors[2], $\phi_u \in \mathbb{R}^k$ for each user u and $\psi_i \in \mathbb{R}^k$ for each item i, and assume a multiplicative model for the dyadic response,

$$p(y_{ui}|u, i) = p(y_{ui}|r_{ui}; \Theta),$$

where Θ denotes the set of hyper-parameters, the utility is assumed as a multiplicative function of the latent factors,

$$r(u, i) = \phi_u^\top \psi_i.$$

This way the factors could explain past responses and in turn make prediction for future ones. This model implicitly encodes the Aldous-Hoover theorem [36] for exchangeable matrices – y_{ui} are independent of each other given ϕ_u and ψ_i. In essence, it amounts to a low-rank approximation of the matrix Y that naturally embeds both users and items into a vector space in which the inner-products directly reflect the semantic relatedness.

To design a concrete model [2, 62, 77], one needs to specify a distribution for the dependence. Afterwards, the model boils down to an optimization problem. For example two commonly-used formulations are:

[1] Note the subtle difference in data representation: while we record entire sessions, CF only records the dyadic responses.

[2] We assume each latent factor ϕ and ψ contains a constant component so as to absorb user/item-specific offset into these factors.

- ℓ_2 **regression** The most popular learning formulation is to minimize the ℓ_2 loss within an empirical risk minimization framework [41]:

$$\min_{\phi,\psi} \sum_{(u,i)\in\Omega} (y_{ui} - \phi_u^\top \psi_i)^2 + \lambda_{\mathcal{U}} \sum_{u\in\mathcal{U}} ||\phi_u||^2 + \lambda_{\mathcal{I}} \sum_{i\in\mathcal{I}} ||\psi_i||^2,$$

where Ω denotes the set of (u, i) dyads for which the responses y_{ui} are observed, $\lambda_{\mathcal{U}}$ and $\lambda_{\mathcal{I}}$ are regularization weights.

- **Logistic** Another popular formulation [62, 1] is to use logistic regression by optimizing the cross-entropy:

$$\min_{\phi,\psi} \sum_{(u,i)\in\Omega} \log\left[1 + \exp(-\phi_u^\top \psi_i)\right] + \lambda_{\mathcal{U}} \sum_{u\in\mathcal{U}} ||\phi_u||^2 + \lambda_{\mathcal{I}} \sum_{i\in\mathcal{I}} ||\psi_i||^2$$

3.3.3 Motivating discussions

Collaborative filtering approaches have made substantial progresses and are currently the state-of-the-art techniques for recommender system. However, we argue here that CF approaches might be a bit lacking in several aspects. First of all, although data sparseness is a big issue, CF does not fully leverage the wealth of user behavior data. Take the user-recommender interaction process described in §3.2 as an example (c.f. Table 1), CF methods typically use only the action dyad (u, i^*) of each session while other dyads $\{(u, i)|i \in \mathcal{O}, i \neq i^*\}$ are treated missing and totally disregarded, which could be wasteful of the invaluable training resource because these non-action dyads are not totally useless, as shown by the experiments in this chapter.

Secondly, most existing CF approaches learn user preference collaboratively by either approximating the dyadic responses $\{y_{ui^*}\}$ [72, 60, 1, 41] or preserving the ordinal ranking information derived from the dyadic responses [84, 51]; none of them models the user choice behavior. Particularly, as users choose from competing alternatives, there is naturally a local competition effect among items being offered in a session. Our work show that this effect could be an important clue for learning user preference.

Because latent factor models are very flexible and could be under-determined (or over-parameterized) even for rather moderate number of users/items. With the above two limitations, CF approaches are vulnerable to over-fitting [1, 41]. Particularly, while most existing CF models might learn consistently on user ratings (numerical value typically with five levels) if given enough training data, they usually perform poorly on binary responses. For example, for the aforementioned interaction process (*c.f.* Table 1), the response y_{ui} is typically a binary event indicating whether or not item i was accepted by the user u. With the non-action dyads being ignored, the responses are exclusively positive observations (either $y_{ui} = 1$ or missing). As a result, we will obtain an overly-optimistic estimator that biases toward positive responses and predicts positive for almost all the incoming dyads (See §3.8.1 for empirical evidences).

3.4 Local optimality of user choices

In this section, we present some axiomatic views of the user choice behaviors.

Formally, the individual choice process (i.e. user-recommender interactions) in a recommender system can be viewed as an instance of the *opportunity give-and-take* (GAT) process.

Definition [GAT]: *An opportunity give-and-take process is a process of interactions among an agent u, a system S and a set of opportunities $\mathcal{I}$; at an interaction t:*

- *u is given a set of opportunities $O_t \subset \mathcal{I}$ by S;*
- *u makes the decision by takeing one of the opportunities: $i_t^* \in \mathcal{O}_t$;*
- *Each opportunity $i \in \mathcal{O}_t$ could potentially give u a revenue (utility) of r_{ui} if being taken or 0 otherwise.*

Note that we assume the agent is *a priori* not aware of all the items, and only through the recommender S can she get to know the items, therefore other items that are not in $\mathcal{O}_t$ is unaccessible to u at interaction t. This is reasonable considering that the total number of items in the inventory is usually very large. Moreover, we assume an agent u is a rational decision maker: she knows that her choice of item i will be at the expense of others $i' \in \mathcal{O}_t$, therefore she compares among alternatives before making her choice. In other words, for each decision, u considers both revenue and opportunity cost, and decides which opportunity to take based on the potential profit of each opportunity in $\mathcal{O}$. Specifically, the opportunity cost c_{ui} is the potential loss of u from taking an opportunity i that excludes her to take other opportunities: $c_{ui} = \max\{r_{ui'} : i' \in \mathcal{O} \setminus i\}$; the profit $\pi_{ui} = r_{ui} - c_{ui}$ is the net gain of an decision. By drawing the rational decision theory [53], we present the following principle of individual choice behavior.

Proposition: *A rational decision is a decision maximizing the profit: $i^* = \arg\max_{i \in \mathcal{O}} \pi_{ui}$.*

This proposition implies the constraint of "local optimality of user choice", a local competitive effect restricting that the agent u always chooses the offer that is locally optimal in the context of the offer set $\mathcal{O}_t$.

3.5 Collaborative competitive filtering

We present a novel framework for recommender learning by modeling the user-system interaction process. The key insight is that the contexts $\mathcal{O}_t$ in which user's decisions are made should be taken into account when learning recommender models. In practice, a user u could make different decisions when facing different contexts $\mathcal{O}_t$. For instance, an item i would not have been chosen by u if it were not presented to her at the first place; likewise, user u could choose another item if the context $\mathcal{O}_t$ changes such that a better offer (e.g., a more interesting item) is presented to her.

The local-optimality principle induces a constraint which could be translated to an objective function for recommender learning:

$$\begin{aligned} &\forall i^* \in \mathcal{D}_t, \quad r_{ui^*} \geqslant \max\{r_{ui} | i \in \mathcal{O}_t \setminus \mathcal{D}_t\} \\ \text{or} \quad &P(i^* \text{ is taken}) = P(r_{ui^*} \geqslant \max\{r_{ui} | i \in \mathcal{O}_t \setminus \mathcal{D}_t\}). \end{aligned} \tag{1}$$

This objective is, however, problematic. First, the inequality constraint restricts the utility function only up to an arbitrary order-preserving transformation (e.g. a monotonically increasing function), and hence cannot yield a unique solution (e.g. point estimation) [56]. Second, optimization based on the induced objective is computationally intractable due to the max operator. To this end, we present two surrogate objectives, which are both computationally efficient and show close connections to existing models.

3.5.1 Softmax model

Our first formulation is based on the random utility theory [53, 56] which has been extensively used for modeling choice behavior in economics [59] and marketing science [27]. In particular, we assume the utility function consists of two components $r_{ui} + e_{ui}$, where: (1) r_{ui} is a deterministic function characterizing the intrinsic interest of user u to item i, for which we use the latent factor model to quantify $r_{ui} = \phi_u^\top \psi_i$; (2) the second part e_{ui} is a stochastic error term reflecting the uncertainty and complexness of the choice process[3]. Furthermore, we assume the error term e_{ui} is an independently and identically distributed Weibull (extreme point) variable:

$$P(e_{ui} \leqslant \epsilon) = e^{-e^{-\epsilon}}.$$

Together with the local-optimality principle, these two assumptions yield the *multinomial logit model* [59, 56, 27]:

$$p(i^* = i|u, \mathcal{O}) = \frac{e^{r_{ui}}}{\sum_{j \in \mathcal{O}} e^{r_{uj}}} \text{ for all } i \in \mathcal{O}. \tag{2}$$

Intuitively, this model enforces the local-optimality constraint by using the *softmax* function as a surrogate of *max*.

Given a collection of training interactions $\{(u_t, \mathcal{O}_t, i_t^*)\}$, the latent factors can be estimated using penalized maximum likelihood via

$$\min_{\phi,\psi} \sum_t \log\Big[\sum_{i \in \mathcal{O}_t} \exp(\phi_{u_t}^\top \psi_i)\Big] - \phi_{u_t}^\top \psi_{i_t^*} + \lambda_{\mathcal{U}} \sum_{u \in \mathcal{U}} ||\phi_u||^2 + \lambda_{\mathcal{I}} \sum_{i \in \mathcal{I}} ||\psi_i||^2. \tag{3}$$

While the above formulation is a convex optimization w.r.t. r_{ui} as each of the objective terms in Eq(3) is strongly concave, it is nonconvex w.r.t. the latent factors ϕ and ψ. We postpone the discussion of optimization algorithms to §3.6.

3.5.2 Hinge model

Our second formulation is based on a simple reduction of the local-optimality constraint by noting, from Eq(1), that:

$$\begin{aligned} P(i = i^*|u, \mathcal{O}) &= P((r_{ui^*} - r_{ui}) > (e_{ui} - e_{ui^*}),\ \forall i \in \mathcal{O}) \\ &\leqslant P((r_{ui^*} - \bar{r}_{u\tilde{i}}) > (\bar{e}_{u\tilde{i}} - e_{ui^*})), \end{aligned}$$

where $\bar{r}_{u\tilde{i}} = \frac{1}{|\mathcal{O}|-1} \sum_{i \in \mathcal{O} \setminus i^*} r_{ui}$ is the average potential utility that u could possibly gain from the non-chosen items. Intuitively, the above model encourages that the utility difference between choice and non-chosen items, $r_{ui^*} - \bar{r}_{u\tilde{i}}$, to be nontrivially greater than random errors. Based on this notion, we present the following formulation which views the task as a pairwise preference learning problem [32] and uses the non-choices averagely as negative preferences.

$$\begin{aligned} &\min_{\phi,\psi,\xi} \sum_t \xi_t + \lambda_{\mathcal{U}} \sum_{u \in \mathcal{U}} ||\phi_u||^2 + \lambda_{\mathcal{I}} \sum_{i \in \mathcal{I}} ||\psi_i||^2 \\ &\text{s.t.: } r_{ui_t^*} - \frac{1}{|\mathcal{O}_t| - 1} \sum_{i \in \mathcal{O}_t \setminus \{i_t^*\}} r_{ui} \geq 1 - \xi_t \text{ and } \xi_t \geq 0. \end{aligned} \tag{4}$$

[3]The error term essentially accounts for all the subtle, uncertain and unmeasurable factors that influence user choice behaviors, for example, a user's mood, past experience, or other factors (e.g., whether the decision is made in a hurry, together with her friends, or totally unconsciously)

This formulation is directly related to the maximum score estimation [56] of the multinomial logit model Eq(2). Intuitively, it directly reflects the insight that user decisions are usually made by comparing alternatives and considering the *difference* of potential utilities. In other words, it learns latent factors by maximizing the marginal utility between user choice and the average of non-choices.

Again, the optimization is convex w.r.t. r_{ui}, but nonconvex w.r.t. the latent factors, therefore the standard optimization tools such as the large variety of RankSVM [32] solvers are not directly applicable.

3.5.3 Complexity

It is worth noting that our CCF formulations have an appealing linear complexity, $O(|\mathcal{I}| \times |\mathcal{O}|)$, where the offer size $|\mathcal{O}|$ is typically a very small number. For example, Netflix recommends $|\mathcal{O}| = 7$ movies for each visit, and Yahoo! frontpage highlights $|\mathcal{O}| = 4$ hot news for each browser. Therefore, CCF has the same-order complexity as the rating-oriented CF models. Note that the ranking-oriented CF approaches [84, 51] are much more expensive – for each user u, the learning complexity is quadratic $O(|\mathcal{I}|^2)$ as they learn preference of each user by comparing every pair of the items.

3.6 Learning algorithms

As we have already mentioned, due to the use of bilinear terms, both of the two CCF variants are nonconvex optimization problems regardless of the choice of the loss functions. While there *are* convex reformulations for some settings they tend to be computationally inefficient for large scale problems as they occur in industry — the convex formulations require the manipulation of a full matrix which is impractical for anything beyond thousands of users.

Moreover, the interactions between user and items change over time and it is desirable to have algorithms which process this information incrementally. This calls for learning algorithms that are sufficiently efficient and preferably capable to update dynamically so as to reflect upcoming data streams, therefore excluding offline learning algorithms such as classical SVD-based factorization algorithms [41] or spectral eigenvalue decomposition methods [51] that involve large-scale matrices.

We use a distributed stochastic gradient variant with averaging based on the Hadoop MapReduce framework. The basic idea is to decompose the objectives in Eq(3) or Eq(4) by running stochastic optimization on sub-blocks of the interaction traces in parallel in the Map phase, and to combine the results for latent factors in the Reduce phase.

3.6.1 Stochastic optimization

We derive a stochastic gradient descent algorithm to solve the optimization described in Eq(3) or Eq(4). The algorithm is computationally efficient and decouplable among different interactions and users, therefore amenable for parallel implementation.

The algorithm loops over all the observations and updates the parameters by moving in the direction defined by negative gradient. Specifically, we can carry out the following update equations on each machine separately:

- For all $i \in \mathcal{O}_t$ do $\psi_i \leftarrow \psi_i - \eta \left[l'(\phi_u^\top \psi_i)\phi_u + \lambda_{\mathcal{I}} \psi_i \right]$.
- For each u do $\phi_u \leftarrow \phi_u - \eta \left[\sum_{i \in \mathcal{O}_t} l'(\phi_u^\top \psi_i)\psi_i + \lambda_{\mathcal{U}} \phi_u \right]$.

Here η is the learning rate[4]. The gradients are given by:

$$l'_{\text{Softmax}}(r_{ui}) = \frac{\exp(r_{ui})}{\sum_{j\in\mathcal{O}}\exp(r_{uj})} - \delta_{i,i^*} \tag{5}$$

$$l'_{\text{Hinge}}(r_{ui}) = -\frac{|\mathcal{O}|\,\delta_{i,i^*} - 1}{|\mathcal{O}| - 1} H(1 - r_{ui^*} + \bar{r}_{u\tilde{i}}) \tag{6}$$

where $H(\cdot)$ is the Heaviside function, i.e. $H(x) = 1$ if $x > 0$ and $H(x) = 0$ otherwise.[5]

3.6.2 Feature hashing

A key challenge in learning CCF models on large-scale data is that the storage of parameters as well as observable features requires a large amount of memory and a reverse index to map user IDs to memory locations. In particular in recommender systems with hundreds of millions of users the memory requirement would easily exceed what is available on today's computers (100 million users with 100 latent feature dimensions each amounts to 40GB of RAM). We address this problem by implementing feature hashing [85] on the space of matrix elements. In particularly, by allowing random collisions and applying hash mapping to the latent factors (i.e. ϕ and ψ), we keep the entire representation in memory, thus greatly accelerating optimization.

3.7 Extensions

We now discuss two extensions of CCF to address the fact that in some cases users choose not to respond to an offer at all and that moreover we may have observed features in addition to the latent representation discussed so far.

3.7.1 Sessions without response

In establishing the CCF framework for modeling the user choice behavioral data, we assumed that for each user-system interaction t, the decision set $\mathcal{D}_t$ contains at least one item. This assumption is, however, not true in practice. A user's visit at a recommender system does not always yield an action. For example, users frequently visit online e-commerce website without making any purchase, or browse a news portal without clicking on any advertisement. Actually, such nonresponded visits may account for a vast majority of the traffics that an recommender system receives. Moreover, different users may have different propensities for taking an action. Here, we extend the multinomial logit model to modeling both responded and nonresponded interactions, $(u_t, \mathcal{O}_t, i_t^*)$ and $(u_t, \mathcal{O}_t, \emptyset)$ respectively.

This is accomplished by adding a scalar θ_u for each user u to capture the *action threshold* of user u. We assume that, at an interaction t, user u_t takes an effective action only if she feels that the overall quality of the offers $\mathcal{O}_t$ are good enough and worth the spending of her attention. In keeping with the multinomial logit model this means that

$$p(i^* = i|u, \mathcal{O}) = \frac{\exp(\phi_u^\top \psi_i)}{\exp(\theta_u) + \sum_{j\in\mathcal{O}}\exp(\phi_u^\top \psi_j)} \tag{7}$$

[4]We carry out an annealing procedure to discount η after each iteration, as suggested by [41].

[5]We approximate this by the continuous function $\frac{1}{1+e^{-100x}}$, which helps with convergence.

for all $i \in \mathcal{O}$ and the probability of no response is given by the remainder, that is by:

$$\frac{\exp(\theta_u)}{\exp(\theta_u) + \sum_{j \in \mathcal{O}} \exp(\phi_u^\top \psi_j)}.$$

In essence, this amounts to a model where the 'non-response' has a certain reserve utility that needs to be exceeded for a user to respond. We may extend the hinge model in the same spirit (we use a trade-off constant $C > 0$ to calibrate the importance of the non-responses):

$$\begin{aligned} \min_{\phi,\psi,\xi,\varepsilon,\theta} \ & \sum_t \xi_t + C \sum_t \varepsilon_t + \lambda_{\mathcal{U}} \sum_{u \in \mathcal{U}} ||\phi_u||^2 + \lambda_{\mathcal{I}} \sum_{i \in \mathcal{I}} ||\psi_i||^2 \\ \text{subject to } & r_{u_t i_t^*} - \bar{r}_{u_t \tilde{i}} - \theta_u \geqslant 1 - \xi_t \text{ for all } i^* \in \mathcal{D}_t \text{ if } \mathcal{D}_t \neq \emptyset \\ & \theta_u - r_{u_t i} \geqslant 1 - \varepsilon_t, \quad \forall i \in \mathcal{O}_t \text{ if } \mathcal{D}_t = \emptyset \\ & r_{ui} = \phi_u^\top \psi_i \text{ and } \xi_t \geqslant 0 \text{ and } \varepsilon_t \geqslant 0 \end{aligned} \tag{8}$$

3.7.2 Content features

In previous sections, we use a plain latent factor model for quantifying utility, i.e. $r_{ui} = \phi_u^\top \psi_i$. A known drawback [1] of such model is that it only captures dyadic data (responses), and therefore generalizes poorly for completely new entities, i.e. unseen users or items, of which the observations are missing at the training stage. Here, we extend the model by incorporating content features. In particular, we assume that, in addition to the latent features ϕs and ψs, there exist some observable properties $x_u \in \mathbb{R}^m$ (e.g. a user's self-crafted registration files) for each user u, and $x_i \in \mathbb{R}^n$ (e.g. a textual description of an item) for each item i. We then assume the utility r_{ui} as a function of both types of features (i.e. observable and latent) [93]:

$$r_{ui} \sim p(r_{ui} | \phi_u^\top \psi_i + x_u^\top M x_i)$$

where the matrix $M \in \mathbb{R}^{m \times n}$ provides a bilinear form for characterizing the utility based on the content features of the corresponding dyads. This model integrates both collaborative filtering [41] and content filtering [17]. On the one hand, if the user u or item i has no or merely non-informative observable features, the model degrades to a factorization-style utility model. On the other hand, if we assume that ϕ_u and ψ_i are irrelevant, for instance, if i or j is totally new to the system such that there is no interaction involving either of them as in a cold-start setting [104], this model becomes the classical content-based relevance model commonly used in, e.g. webpage ranking [102], advertisement targeting [16], and content recommendation [17].

3.8 Experiments

We report experimental results on two test-beds. First, we evaluate the CCF models with CF baselines on two dyadic data sets with simulated choice contexts. The choice of simulated data generated from CF datasets was made since we are unaware of any *publicly* available datasets directly suitable for CCF. Furthermore, we extend our evaluation to a more strict setting based on user-system interaction session data from a commercial recommender system.

Table 2: Statistics of the three recommendation data sets.

	#user	#item	#dyads	offer size
Social	1.2M	400	29M	-
Netflix-5star	0.48M	18K	100M	-
News	3.6M	2.5K	110M	4

3.8.1 Dyadic response data

We use dyadic data with binary responses, i.e. $\{(u, i, y_{ui})\}$ where $y_{ui} \in \{1, \text{missing}\}$. We compare different recommender models in terms of their top-k ranking performance.

Social network data. The first data set we used was collected from a commercial social network site, where a user expresses her preference for an item with an explicit indication of "like". We examine data collected for about one year, involving hundreds of millions of users and a large collection of applications, such as games, sports, news feeds, finance, entertainment, travel, shopping, and local information services. Our evaluation focuses on a random subset consisting of about 400 items, 1.2 million users and 29 million dyadic responses ("like" indications).

Netflix 5 star data. We also report results on a data set derived from the Netflix prize data[6], one of the most famous public data sets for recommendation. The Netflix data set contains 480K users and 18K movies. We derive binary responses by considering only 5-star ratings as "positive" dyads and treating all the others as missing entries.

For both data sets, we randomly split the data into three pieces, one for training, one for testing and the other for validation.

Evaluation metrics. We assess the recommendation performance of each model by comparing the top suggestions of the model to the true actions taken by a user (i.e. "like" or 5-star). We consider three measures commonly used for accessing top-k ranking performance in the IR community:

AP is the *average precision*. AP@n averages the precision of the top-n ranked list of each query (e.g. user).

AR or *average recall* is the average recall of the top-n rank list of each query.

nDCG or *normalized Discounted Cumulative Gain* is the normalized position-discounted precision score. It gives larger credit to top positions.

For all the three metrics we use $n = 5$ since most social networks and movie recommendation sites recommend a similar number of items for each user visit.

Evaluation protocol. We compare the two CCF models (i.e. Softmax and Hinge) with the two standard CF factorization models (i.e. ℓ_2 and Logistic) described in §3.3. For dyadic data with binary responses, the Logistic CF model amounts to the state-of-the-art [62, 1].

We adopt a fairly strict top-k ranking evaluation. For each user, we assess the top results out of a total preference ordering of the whole item set. In particular, for each user u, we consider all the items as candidates; we compute the three measures based on the comparison between the ground truth (the set of items in the test set that user u actually liked) and the top-5 suggestions predicted by each model. For statistical consistency, we employ a cross-validation style procedure. We learn the models on training data with

[6] `http://www.netflixprize.com`

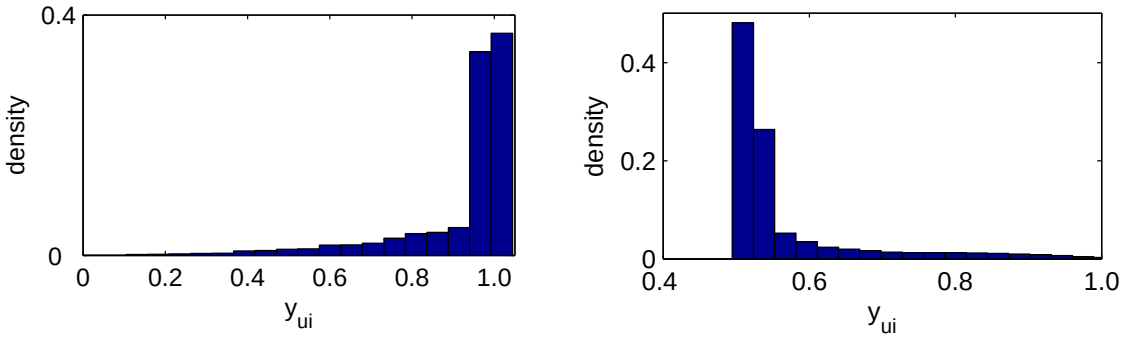


Figure 2: Histograms of the predicted dyadic responses obtained by CF and CCF.

Table 3: Comparison of top-k ranking performance on the two dyadic data sets with simulated contexts.

Model		AP@5	AR@5	nDCG@5
Social				
CF	ℓ_2	0.448	0.230	0.475
CF	Logistic	0.449	0.230	0.476
CCF	Softmax	**0.688**	**0.261**	**0.704**
CCF	Hinge	0.686	0.260	0.702
Netflix-5star				
CF	ℓ_2	0.135	0.022	0.145
CF	Logistic	0.135	0.023	0.146
CCF	Softmax	**0.186**	**0.033**	**0.189**
CCF	Hinge	0.185	0.032	0.188

parameters tuned on validation data, and then apply the trained models to the test data to assess the performance. All three measures reported are computed on test data only, and they are averaged over five random repeats (i.e. random splits of the data).[7]

To render the data compatible with CCF we simulate a fixed-size pseudo-offer set for each interaction. Specifically, in each step of the stochastic optimization at a positive observation, e.g. $y_{ui} = 1$, we randomly sample a handful set of missing (unobserved) entries $\{y_{ui'}\}_{i'=1:m}$. These sampled dyads are then treated as non-choices, and together with the positive dyad, they are used as the offer set for the current session. In our experiments, we choose $m = 9$ pseudo non-choices; in other words, we assume the offer size $|\mathcal{O}_t|$ =10.

Results and analysis. We report the mean scores in Table 3. Since the dataset are fairly large the standard deviations of all values were below 0.001. Consequently we omitted the latter from the results. As can be seen from the table, CCF dramatically outperforms CF baselines on both data sets. In terms of AP@5, the two CCF models gain about 52.8%–53.6% improvements compared to the two CF models on the Social data, and by 37.0%–37.8% on Netflix-5 star. Similar comparisons apply to the nDCG@5 measure. And in terms of the AR@5, CCF models outperform CF competitors by up to 13.5% on Social,

[7] *Note that the contextual information (the offer set $\mathcal{O}_t$ for each interaction t) is missing for both of the two dyadic data sets.* We use the datasets with simulated contexts. Results on real interaction data are reported in §3.8.2.

and 30% on Netflix-5 star data. All these improvements are statistically highly significant. Note that these results are quite consistent: both CF models perform comparably with each other on both data sets; the performance of the two CCF variants is also comparable; between the two groups, gaps are noticeable.

One argument we made in this chapter for motivating our work is that since the CF models disregard the context information and only learns on positive (action) dyads, they almost inevitably yield overly-optimistic predictions (i.e. predicting positive for all possible dyads). We hypothesize that such estimation *bias* is one of the key reasons for the inability of CF models in learning binary dyadic data. As an empirical validation, in Figure 2, we plot the histograms of the predicted dyadic responses $\hat{y}_{ui}$ (i.e. entries of the diffused matrices) obtained by a CF model (ℓ_2) and a CCF model respectively.[8] As we can see, the CF model indeed predicts "positive" for most (if not all) dyads; in contrast, the results obtained by the CCF model demonstrate a more realistic power-law distribution [21].[9]

In reality, since the total number of items in the inventory is too large, each user can only afford to "like" a few items out of the huge amount of alternatives. This power-law property is crucial for information filtering because we are intended to identify a few truly relevant items by *filtering* out many many irrelevant ones. A power-law recommender is desirable in a way analogous to a filter with narrow-bandwidth, which effectively filters out the noises (i.e. irrelevant items) and only let the true signal (i.e. relevant items) pass to the users.

3.8.2 User-system interaction data

We now move on to a more realistic evaluation by applying CCF to real user-system interaction data. We evaluate CCF in both an offline test and an online test while comparing its results to both CF baselines.

Data. We collected a large-scale set of user-system interaction traces from a commercial News article recommender system. In each interaction, the system offers four personalized articles to the visiting user, and the user chooses one of them by clicking to read that article. The recommendations are dynamically changing over time even during the user's visit. The system regularly logs every click event of every user visit. It also records the articles being presented to users at a series of discrete time points. To obtain a context set for each user-system interaction, we therefore trace back to the closest recording time point right before the user-click, and we use the articles presented at that time point as the offer set for the current session. We collected such interaction traces from logged records of over one month. We use a random subset containing 3.6 million users, 2500 items and over 110 million interaction traces. Learning an effective recommender on this data set is particularly challenging as the article pool is dynamically refreshing, and each article only has a lifetime of several hours — it only appears once within a particular day, is pulled out from the pool afterward and never appears again.

Evaluation protocol. We consider the following two evaluation settings, one offline and the other offline.

[8] Similar results obtained with other losses.

[9] Note that the distribution starts at around 0.5 instead of 0, which is consistent with our intuitions that there is actually *no* truly "irrelevant" item for a user – any item has potential utility for a user; user choose one over another based on the relative preference rather than absolute utility. This is true especially in this era of information explosion, where a user is typically facing so many alternatives that she can only pick the one she likes *the most* while ignoring the others.

Table 4: Offline test (top-k ranking performance) on user-system interaction data.

Model		AP@4	AR@4	nDCG@4
30% Training				
CF	ℓ_2	0.245	0.261	0.255
CF	Logistic	0.246	0.263	0.257
CCF	Softmax	**0.262**	**0.278**	**0.274**
CCF	Hinge	0.261	**0.278**	0.273
50% Training				
CF	ℓ_2	0.250	0.273	0.268
CF	Logistic	0.252	0.276	0.269
CCF	Softmax	**0.266**	**0.285**	**0.278**
CCF	Hinge	0.265	**0.285**	0.277
70% Training				
CF	ℓ_2	0.253	0.275	0.271
CF	Logistic	0.253	0.276	0.274
CCF	Softmax	**0.267**	**0.287**	**0.280**
CCF	Hinge	**0.267**	0.286	**0.280**

Offline evaluation Similar to the evaluations presented in §3.8.1, we evaluate the learned recommender models in terms of the top-k ranking performance on a hold-out test subset. We follow the same configurations in §3.8.1 and use the three ranking measures, i.e. AP@n, AR@n and nDCG@n as the evaluation metrics. Note that here we use $n = 4$ instead of 5, because it is the default recommendation size used in the news recommender system.

Online evaluation We further conduct an online test. In particular. for each incoming interaction, we use the trained models to predict which item among the four recommendations will be taken by the user. This prediction is of crucial importance because one of the key objectives for a recommender system is to maximize the traffic and monetary revenue by lifting the click-through rate.

Offline test results. In this setting, we train each model on progressive proportions of 30%, 50% and 70% randomly-sampled training data respectively, and evaluate each trained model in terms of offline top-k ranking performance. The results are reported in Table 4. The two CCF models greatly outperform the two CF baselines in all the three evaluation metrics. Specifically, CCF models gain up to 6.9% improvement over the two CF models in terms of average precision; up to 6.5% in terms of average recall, and up to 7.5% in terms of nDCG. We also conducted a t-test with a standard 0.05 significance level, which further indicate that all the improvements obtained by CCF are significant.

It is worth noting that the improvements obtained by CCF compared to CF baselines are especially evident when the training data are sparser (e.g. using only 30% of training data). This observation empirically validates our argument that the contexts contain substantial useful information for learning recommender models especially when the dyadic action responses are scarce.

The offline results obtained by CCF are quite satisfactory. For example, the average precision is up to 0.276, which means, out of the four recommended items, on average 1.1 are truly "relevant" (i.e. actually being clicked by the user). This performance is quite

Table 5: Online test (predicted click probability) on user-system interaction data.

Model		30%train	50%train	70%train
Random		0.250		
CF	ℓ_2	0.337	0.343	0.347
CF	Logistic	0.341	0.345	0.347
CCF	Softmax	0.376	0.384	**0.391**
CCF	Hinge	**0.377**	**0.385**	**0.391**

promising especially considering that most of the articles in the content pool are transient and subject to dynamically updating.

Online test results. We further evaluate the online performance of each compared model by assessing the predicted click rates. Click-rate is essential for an online recommender system because it is closely-related to both the traffic and the revenue of a webshop. In our evaluation, for each of the incoming visits $(u_t, \mathcal{O}_t, i_t^*)$, we use the trained models to predict the user choice, i.e. we ask the question: "among all the offered items $i \in \mathcal{O}_t$, which one will most likely be clicked?" We use the trained model to rank the items in the offer set, and compare the top-ranked item with the item that was actually taken (i.e. i_t^*) by user u_t. We evaluate the results in terms of the prediction accuracy.

The results are given in Table 5. Because the size of each offer set in the current data set is 4, a random predictor yields 0.25. As seen from the table, while all the four models obtain significantly better predictions than the random predictor, the two CCF models further greatly outperform the two CF models. Specifically, we observe 11.3%–12.7% improvements obtained by CCF models compared to the two CF competitors. These results are quite significant especially considering the dynamic property of the system.

Impact of parameters. The performance of the two CCF models is affected by the parameter settings of the latent dimensionality, k, as well as the regularization weights, $\lambda_{\mathcal{I}}$ and $\lambda_{\mathcal{U}}$. In Figure 3[10], we illustrate how the offline top-k ranking performance changes as a function of these parameters, where we use the same value for both $\lambda_{\mathcal{I}}$ and $\lambda_{\mathcal{U}}$. Here we only reported the results with nDCG@5 measure because the results show similar shapes when other measures (including the click rate) are used. As can be seen from the Figure, the nDCG curves are typically in the inverted U-shape with the optimal values achieved at the middle. In particular, for both the two CCF models, the dimensionality around 20 and regularization weight around 0.0001 yield the best performance, which is also the default parameter setting we used in obtaining our reported results.

Nonresponded sessions. In Section 3.7 we presented two models for encoding nonresponded interactions, e.g. a user visits the News website but does not click any of the recommended articles. These approaches are promising because compared to the responded sessions, the nonresponded ones are typically much more plentiful and if learned successfully, this wealth of information has a potential to alleviating the critical data-sparse issue in recommendation.

Unfortunately, due to the data-logging mechanism of the News recommender system, we were unable to obtain such nonresponded interactions. Instead, for a preliminary test, we conducted evaluation on a small set of *pseudo* nonresponded sessions that are derived

[10]Due to heavy computational consumptions, these results are obtained on a relatively small subset of data.

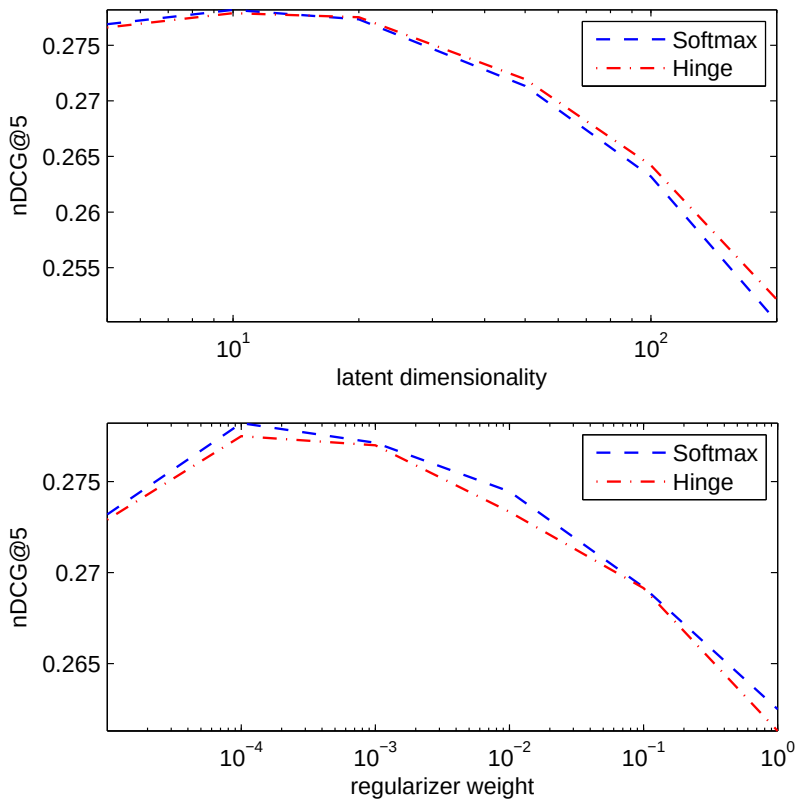


Figure 3: Offline top-k ranking performance as a function of latent dimensionality and regularization weight.

from the responded ones. In particular, we hold out a randomly-sampled subset of sessions; for each of these sessions, we hide the item being clicked by the user, and use the remaining items as a nonresponded context set by assuming no click for this set. We augmented this set of derived nonresponded sessions to the training set, and train the model on the combined training data. The results from this preliminary evaluation did not show significant performance improvement. This is likely due to the fact that the surrogate distribution is invalid. A detailed analysis with more realistic data is the subject of future research.

3.9 Related Work

Although a natural reflection of a user's preference is the process of interaction with the recommender, to our knowledge, this interaction data has not been exploited for learning recommender models. Instead, research on recommender systems has focused almost exclusively on learning the dyadic data. Particularly collaborative filtering approaches only capture the user-item dyadic data with explicit user actions while the context dyads are typically treated missing values. For example, the rating-oriented models aim to approximating the ratings that users assigned to items [72, 60, 1, 41]; whereas the recently proposed ranking-oriented algorithms [84, 51] attempt to recover the ordinal ranking information derived from the ratings.

By exploiting past records of user-item dyadic responses for future prediction based on either *neighborhood based* [72, 60, 51] or *latent factor based* methods [1, 41, 84], collaborative

filtering approaches encode the collaboration effect that similar users get similar preference on similar items. In this chapter, by leveraging the user-recommender interaction data, we show that much better recommender performance can be obtained when a local-competition effect underlying the user choice behaviors is also encoded.

The multinomial logit model we present is derived based on the random utility theory [53, 56]. The model is well-established and has been widely used for a long time in, e.g. psychology [53], economics [59, 56] and marketing science [27]. Particularly, [27] applied the model to examine the brand choice of households on grocery data; [24] showed this model is theoretically and empirically superior to the ℓ_2 regression model. More recently, the pioneering work of [22] first applied the model to characterize online choices in recommender system and investigated how recommender systems impact sales diversity. Following these steps, this work further employs the model to learn factorization models for recommendation.

The Hinge formulation of CCF shows close connection to the pairwise preference learning approaches widely used in Web search ranking [32]. Our model, however, differs from these content filtering models [32] in that instead of learning a feature mapping as in [32], our model uses the formulation for learning a multiplicative latent factor model.

3.10 Summary

In this chapter, we proposed a framework for learning user preference by modeling user choice behavior in the user-system interaction process. Instead of modeling only the sparse binary events of user actions as in traditional collaborative filtering, the proposed *collaborative-competitive filtering* models take into account the contexts in which user decisions are made. We presented two models in this spirit, established efficient learning algorithms and demonstrated the effectiveness of the proposed approaches with extensive experiments on three large-scale real-world recommendation data sets.

A key feature of the current chapter is that we assume the recommender adopts a predefined and fixed strategic policy when making recommendations, e.g., via simply preference based ranking. In the next chapter, we show that a recommender system could act more strategically to optimize its acts in respect of certain business objectives, which is made possible by a novel game-theoretic formulation for recommendation.

CHAPTER IV

COLLABORATIVE COMPETITIVE FILTERING II

Recommender systems have emerged as a new weapon to help online firms to realize many of their strategic goals (e.g., to improve sales, revenue, customer experience etc.). However, many existing techniques commonly approach these goals by seeking to recover preference (e.g., estimating ratings) in a matrix completion framework. This chapter aims to bridge this significant gap between the clearly-defined strategic objectives and the not-so-well-justified proxy [97]. We show it is advantageous to think of a recommender system as an analogy to a *monopoly economic market* with the system as the sole *seller*, users as the *buyers* and items as the *goods*. This new perspective motivates a game-theoretic formulation for recommendation. In this spirit, we revisit the collaborative competitive filtering (CCF) preference models presented in Chapter 3 and extend it to a game-theoretic framework consisting of two components. The first component is a model for characterizing a buyer u's reaction R to any given action A of the seller, i.e., the conditional distribution $p_u(R|A)$, which is essentially the *multinomial logit* preference models we already described in Chapter 3; The second one is a model for optimizing a recommender's action policy in respect of certain strategic goals. In this chapter, we formally describe the game-theoretic formulation, briefly revisit the $p_u(R|A)$ model and present in detail the formulation for action optimization. We show how objectives such as click-trough rate, sales revenue and consumption diversity can be optimized explicitly in this framework [97].

4.1 Introduction

Recommender systems have become a core component for today's online businesses. With the abilities of connecting merchant *supply* (i.e., items of various types such as retailing products, movies, articles, ads, experts, etc.) to market *demands* (i.e., potentially interested consumers), recommender systems are helping online firms (e.g. Amazon, Netflix, Yahoo!) to realize many of their hard-to-attain business goals (e.g., to boost sales, improve revenue, enhance customer experiences) [12, 11, 22, 79]. Compared to an offline market, online recommender system has the unbeatable convenience in control, intervention, monitoring and measurement of the market, and consequently the appealing opportunity to adjust its operational actions to optimize certain strategic objectives. Surprisingly, despite the fact that many of these goals are clearly defined, they are not reflected in existing techniques for recommendation in a well-justified way. Instead, research on recommendation has been focused almost exclusively on learning preference (e.g., estimating a user's rating to a movie) in a matrix completion formulation [72, 60, 41, 16, 84]. It is rather unclear how preference learning, as a proxy, approximates these goals, or how a strategic intervention should be designed to achieve certain goals.

In this chapter, we seek to bridge this significant gap. We show it is advantageous to look at the *user-system interactions* and think of a recommender system as an analogy to a *monopoly market* in economics (i.e., with the system as the sole seller, users as buyers and items as goods)[1], rather than at *user-item interactions* as in *matrix completion*. This new

[1]Hereafter, we will use interchangeably "system" and "seller", "user" and "buyer", "item" and "good".

perspective motivates a novel game-theoretic formulation, upon which recommendations can be optimized strategically with respect to business objectives such as click-through rate, sales revenue and consumption diversity.

4.1.1 User-System Interactions

Recommender systems are commonly designed by analyzing the dyadic *user-item interactions* as can be recorded by a matrix, for example, users assigning ratings to movies. Research has thus been focused exclusively on estimating preference or equivalently completing the matrix [5, 72, 60, 84, 2, 1, 41, 16]. This *matrix-completion* formulation of recommendation has been extensively investigated and become especially popular thanks to the Netflix Prize Competition. Nonetheless, as we show in this chapter, there are plenty of reasons to look at *user-system interactions* instead and formulate recommendation as *games* [49].

The user-system interactions in a recommender system are a process where the system *acts* per user inquiry and the user *reacts* in response. The most common type of such action-reaction interactions, as are prevalent in today's commercial recommender systems (e.g., Facebook's News feed, Amazon's product recommender, Yahoo! Frontpage), are described below, where at each interaction:

1) a user u inquires the system (e.g. visits a movie recommendation web site);
2) the system *acts* by recommending a set of customized items (i.e. movies of potential interest) to the user;
3) the user *reacts* by choosing and consuming some of the recommended items (e.g. click a link, rent a movie, view a News article, purchase a product).

This process in many aspects resembles what happens in a monopoly market where the recommender system, as the sole seller, has absolute market power to manipulate the market, yet its gain is defined based on the reaction of the buyers (i.e., users), e.g., the success of an advertising system is directly related to how users react (i.e., whether they click the ads or not). This analogy reveals the fundamental insight that *recommendation is not just about preference estimation and should not be formulated as matrix completion*:

1. Preference is not the only factor that determines a buyer's decision, which also depends crucially on the seller's action (i.e., what the seller provides to him).
2. Even for estimating preference, a matrix-completion is suboptimal as we have shown in Chapter 3. Particularly, the seller's action serves as the context in which buyer's decision is made, which is vital for estimating finer preference, yet ignored by matrix-completion.
3. More importantly, although the strategic objectives for a recommender system to achieve are clearly defined, the role of preference estimation as a proxy is not well justified. This is particularly frustrating to practitioners as how a recommender can be optimized w.r.t. certain strategic goals is rather unclear.

For recommender systems to behave more strategically and intelligently, new formulations are therefore needed.

4.1.2 Recommendation as Collaborative Games

In this chapter, we present a game-theoretic formulation for recommendation, where the user-system interactions are modeled as a collection of games with each game played between

one seller and one buyer (i.e., between the system and one user). For the sake of statistical inference, it is nonetheless important *not* to model these games as mutually independent. We therefore bring forward the notion of "collaborative games" that *similar games are expected to yield similar outcomes*, which enables us to pool the sparse data across games to obtain reliable statistical estimation.

The proposed *Collaborative-competitive filtering* (CCF) framework consists of two components: (1) a model for $p_u(R|A)$, which characterizes the reaction R of a buyer u in the context of any given action A of the seller; this conditional distribution characterizes users' behavior and enables us to predict the outcome of a game in advance; and (2) given $p_u(R|A)$ for every buyer u, a formulation for optimizing the seller's action A w.r.t. a predefined payoff (e.g., a strategic goal).

To effectively model $p_u(R|A)$, we establish a single unified model that integrates two distinct methodologies, i.e., latent factor models in collaborative filtering and choice models in econometrics. First, by using latent factor based utility parametrization, the model encodes the "collaboration effects" [14, 61] among games in light of the notion of "collaborative games". As the policy spaces are prohibitively large yet the observations are extremely sparse, this step is essential for reliable statistical inference as it seeks to pool data across games [72, 60, 1, 41]. Second, by drawing on axiomatic perspectives of human behavior, the model quantifies reaction based on the random utility theory in psychology [53] and econometrics [59, 56, 24, 27]. This new formulation is flexible and powerful, and it also remarkably reduces the parametric complexity of $p_u(R|A)$ significantly from prohibitive high-order polynomial scale down to linear scale. We show that important factors such as reaction propensities and position bias can be naturally addressed by this model as well.

The knowledge about users' reaction behavior, as characterized by $p_u(R|A)$, enables us to optimize the action (i.e., recommendation) of the recommender system strategically [49]. For any input action A, the possible outcomes of the games occur with probabilities defined upon $p_u(R|A)$. Given a payoff (i.e., a function of the outcome) that is von Neumann-Morgenstern rational, the expected utility theory asserts that the best action is the one that maximizes the expected payoff [64]. We show how business objectives such as click-trough rate, sales revenue and consumption diversity can be formulated explicitly as expected utilities and used in turn to optimize a recommender system's action strategy.

Interestingly enough, we also show that the CCF model defined above is sequentially rational and thus achieves the *perfect Nash equilibrium* [49]. Experiments on a real-world commercial system demonstrate that the proposed CCF model not only outperforms CF models in both offline and online tests but is also highly effective in achieving satisfactory strategic goals.

4.2 User-Item Interactions and Collaborative Filtering

Many existing approaches generally think of recommendation as *user-item interactions* and therefore aim to recover/estimate the preference of each individual user to the items. Given a set of N users

$$u \in \mathcal{U} := \{1, 2, \ldots, N\}$$

and a set of M items

$$i \in \mathcal{I} := \{1, 2, \ldots, M\},$$

this is naturally formulated as a matrix completion problem, where we are given observations of dyadic responses $\{(u, i, y_{ui})\}$ with each y_{ui} being an observed response indicating user's

preference (e.g. user's rating to an item, or indication of whether user u likes item i), the goal is to complete the whole mapping:

$$(u, i) \rightarrow y_{ui} \text{ where } u \in \mathcal{U}, i \in \mathcal{I}$$

which constitutes a large matrix $Y \in \mathcal{Y}^{|\mathcal{U}| \times |\mathcal{I}|}$. Assume each item can be consumed multiple times, recommendations are usually done by a simple preference-based ranking according to Y, (i.e., recommending the items with highest y_{ui} scores to user u). This formulation include both of the two major categories of approaches to recommendation, i.e., content-based filtering [5, 16] and collaborative filtering [72, 60, 1, 41], as we have already reviewed in Chapter 2.1 and 3.3.

4.3 User-System Interaction as Collaborative Games

Based on the perspective of u*ser-item interactions*, the matrix completion formulation for recommendation has led to numerous algorithms which excel at a number of data sets, including the prize-winning work of [41] and many other successful collaborative filtering algorithms [72, 60, 71, 1, 41, 84, 51]. However, as we mentioned, this formulation is inherently flawed; instead, it is advantageous to consider recommendation rather as *user-system interactions*. Firstly, by considering user-system interaction and putting user's decision into context, we can leverage data that was previously lumped together with missing data in existing CF models and consequently capture more accurate information about user preference and/or reaction behavior. More importantly, it motivates a novel game-theoretic formulation for recommendation and opens up a promising direction, i.e., to optimize recommender systems strategically in respect of important business objectives, which cannot be achieved otherwise with the conventional matrix completion formulation.

Consider the user-system interaction in a recommender system: we have N users $u \in \mathcal{U} := \{1, 2, \ldots, N\}$ and M items $i \in \mathcal{I} := \{1, 2, \ldots, M\}$; when a user u visits the site, the system recommends a set of items $A = \{i_1, \ldots, i_l\}$ and u in turn chooses a (possibly empty) subset $R \subseteq A$ for consumption (e.g. buys some of the recommended products). From now on, we refer to A as *action*, and R as *reaction*. For simplicity, we assume each action is fixed-size with a given length, $|A| = l$, and that each reaction is either empty or contains exactly one choice, $|R| = 1$ or 0. Therefore, we have $A \in \mathcal{A} = \mathcal{I}^l$ and $R \in \mathcal{R} \subset \tilde{\mathcal{I}} = \mathcal{I} \cup \{\emptyset\}$. See Table 1 in Chapter 3 for an example interaction and its degradation to a matrix.

Conceptually, the behavior of the recommender system and that of the users are inter-dependent. On the one hand, since people make different decisions when facing different contexts, a user's decision R depends crucially on the action of the system, A, (i.e., what was provided to him). For instance, an item i would not have been chosen by u if it were not presented to him at the first place; likewise, user u could choose another item if the context A changes such that a better item were presented to him. On the other hand, how a recommender system acts also depend on user's behavior, after all, the success of recommendation (i.e., in terms of click-through, revenue, etc.) is defined directly on how users react to it (e.g., purchase a product, click an ad, rent a movie). It is therefore nature to formulate recommendation based on game theory by thinking of the user-system interaction as a monopoly market: the recommender as the sole seller, a user as a buyer and the items as the goods.

Formally, the user-system interactions in a recommender system can be formulated as a set of non-cooperative games $\mathcal{G} = \{G_n = (P_n, \mathcal{Z}_n, U_n),\ n = 1, 2, \ldots, N\}$. For each game G_n, the player set $P_n = \{S, u_n\}$ consists of two players, i.e., the system (i.e., seller) S and

a user (i.e., buyer) u_n; the policy space $\mathcal{Z}_n = \mathcal{A} \times \mathcal{R} \subset \mathcal{I}^l \times \tilde{\mathcal{I}}$ is the set of all possible action-reaction pairs $Z_n = (A_n, R_n)$, where Z is called an outcome and $\mathcal{Z}$ the outcome space; and the utility (i.e., payoff) function $U_n = \{U_S(Z_n), U_u(Z_n)\}$ consists of the system's payoff U_S and the user's payoff U_u. At an interaction t, a user u_t visits the system and the game G_{u_t} is played with outcome $Z_t = (A_t, R_t)$ and utility output $U(A_t, R_t)$. Since the users' behavior is not in our control, our goal in designing a recommender system is to generate a system action (recommendations) $A_{\tilde{t}}$ for an incoming visit $\tilde{t}$ of user $u_{\tilde{t}}$ so as to maximize the system's payoff $U_s(Z_{\tilde{t}})$.

It is important to emphasize that the games in $\mathcal{G}$ should *not* be modeled as independent games. Particularly, since the outcome space can be very large, yet observations are typically sparse, it is practically important to still be able to leverage the *collaboration effect* such that similar games are expected to yield similar outcomes. This way it enables us to pool the sparse evidences across different but similar games and in turn obtain reliable statistical inference, which is otherwise impossible. For this reason, we term the formulation "*collaborative games*" with a slight abuse of terminology.

This game-theoretic formulation provides a novel perspective for recommendation. Particularly, since the strategies of the buyer and the seller are interdependent, to optimize the seller's action, we have to (1) predict the buyer's reaction R to any given action A in advance; and then (2) optimize the seller's action A to maximize U_s.

4.4 Collaborative competitive filtering revisited

In this section, we revisit the collaborative competitive filtering (CCF) preference model an extend it towards a novel game-theoretic framework for recommendation. The CCF framework consists of two components: (1) a model for characterizing the buyer's (i.e., user's) reaction behavior; and (2) given the information of buyer's behavior, a formulation to optimize the seller's (i.e., the recommender system's) action strategy so as to achieve strategic objectives.

4.4.1 Conditional User Reaction Modeling

The first part of the framework is to predict a buyer's choice R in the context of any given action A of the seller's. In a decision environment with imperfect information, this means to quantify the conditional distribution $p_u(R|A)$. The full parametrized version of this distribution requires $O(NM^{l+1})$ free parameters, statistical estimation of which is practically prohibitive since the observations are typically available only at a scale far less than $O(NM)$ (e.g., in matrix completion, usually less than 1% entries are observed). In this section, we establish an effective model of complexity $O(N + M)$ by integrating latent factor models in collaborative filtering and choice models in econometrics.

4.4.1.1 Behavioral Axioms of Choice Process

We first present an axiomatic view of the choice process. The first axiom is the "local optimality of choice" principle we already described in Chapter 3 and restate here.

Axiom 1 [Local optimality of choice]: *A rational decision is a decision maximizing the profit:* $i^* = \arg\max_{i \in A} \pi_{ui}$.

Unfortunately, this axiom is deterministic, although statistical inference is clearly more favorable due to the complicated nature of human behavior. To this end, we draw an

stochastic counterpart of this axiom from the random utility theory [53, 59]:

AXIOM 2 [INDEPENDENCE OF IRRELEVANT ALTERNATIVES]: *For any given context set A, the relative odds of a user u's selecting an item $i \in A$ over another item $j \in A$ should be independent of the presence or absence of any irrelevant items, i.e.,*

$$\frac{p_u(i|\{i,j\})}{p_u(j|\{i,j\})} = \frac{p_u(i|A)}{p_u(j|A)} \tag{9}$$

Besides other merits, this axiom is especially important as it brings the parametric complexity of $p_u(R|A)$ significantly down from $O(NM^{l+1})$ to $O(NM^2)$, although the latter is still practically prohibitive.

4.4.1.2 User Utility Parametrization

In the spirit of the random utility theory [53, 59], we assume the buyer's utility function consists of two components $U_u(i) = r_{ui} + e_{ui}$, where: (1) r_{ui} is a deterministic component characterizing the intrinsic interest of the buyer u to the good i; (2) the second part e_{ui} is a stochastic unobserved error term reflecting the uncertainty, richness and complexity of the choice process. Under very mild conditions, it has been shown that the error terms e_{ui} are independently and identically distributed with the Weibull (extreme point) distribution [29]:

$$P(e_{ui} \leqslant \epsilon) = e^{-e^{-\epsilon}}. \tag{10}$$

Furthermore, to encode the collaborative effect such that the observed evidences could be pooled across similar games, we parametrize the deterministic utilities, r_{ui}, with the multiplicative latent factor model [41, 1]:

$$r_{ui} = \phi_u^\top \psi_i \tag{11}$$

4.4.1.3 The Multinomial Logit Factor Model

The behavioral axiom and the parametrization we proposed together lead to the following theorem.

THEOREM 1: *Suppose the utility function $U_u(i) = r_{ui} + \epsilon_{ui}$, where ϵ are i.i.d. Weibull variables, then the distribution of selecting one item that satisfies Axiom 2 is given by $p_u(i|A) = e^{r_{ui}} / \sum_{j \in A} e^{r_{uj}}$ for any $i \in A$.*

Proof. c.f. [59]. □

The above model is well-known as the *multinomial logit model*, which has been extensively used for modeling conventional offline consumer choice behavior (e.g., choose of occupation, brand, housing) in econometrics [59, 56], sociometrics [53] and marketing science [24, 27]. We, for the first time, adapt it for modeling online game-theoretic interactions in recommender systems. In contrast to the traditional choice models, where the deterministic part of the utility r_{ui} is a linear mapping $w^\top x_{ui}$ of observed features x_{ui} (i.e., measured user and item features), here we employ the multiplicative latent factor parametrization. The formulation proposed hereby seamlessly integrate two distinct methodologies — choice models in econometrics and factorization models in collaborative filtering. This integration is significant because it enables us to model the seller-buyer games *collaboratively*, rather than *independently* as in conventional choice models. That is, it enables us to pool data

across games such that the interactions engaging similar users, similar actions and similar reactions are dealt with similarly. We also associate with each user a scalar latent factor, θ_u, to capture the *response propensity* of the buyer u. In keeping with the multinomial logit model and the latent factor parametrization, we have the following model

$$p_u(R = i|A) = \frac{\exp(\phi_u^\top \psi_i)}{\exp(\theta_u) + \sum_{j \in A} \exp(\phi_u^\top \psi_j)}, \forall\ i \in A; \tag{12}$$

$$p_u(R = \emptyset|A) = \frac{\exp(\theta_u)}{\exp(\theta_u) + \sum_{j \in A} \exp(\phi_u^\top \psi_j)} \text{ otherwise} \tag{13}$$

which we refer to as *multinomial logit factor* or MLF model. Note that, in contrast to CF, this model is able to leverage all the user-item dyads including those that are treated and disregarded as missing values in CF. Also, it is worth noting that this new formulation is able to further reduces the parametric complexity of $p_u(R|A)$ significantly to linear scale, i.e., $O(k(N + M) + N) \approx O(N + M)$, where k is the dimensionality of the latent factor $\phi \in \mathbb{R}^k$ and $\psi \in \mathbb{R}^k$, which is generally a small number (usually up to a few hundreds).

4.4.1.4 Conditional Maximum Likelihood Estimation

Given a collection of training interactions $\{(u_t, A_t, R_t)\}$, the latent factors, ϕ and ψ, can be estimated using penalized conditional maximum likelihood estimation via

$$\min_{\phi,\psi,\theta,b} \sum_t \left(\log[e^{\theta_{u_t}} + \sum_{i \in A_t} e^{\phi_{u_t}^\top \psi_i}] - (1 - \delta_{\emptyset,t}) \phi_{u_t}^\top \psi_{i^*_{p_t}} - \delta_{\emptyset,t} \theta_{u_t} \right) + \lambda_{\mathcal{U}} \sum_{u \in \mathcal{U}} ||\phi_u||^2 + \lambda_{\mathcal{I}} \sum_{i \in \mathcal{I}} ||\psi_i||^2. \tag{14}$$

where $\delta_{\emptyset,t} = 1$ if $R_t = \emptyset$, or 0 otherwise.

4.4.2 Strategic System Action Optimization

The distribution $p_u(R|A)$ characterizes the buyer's reaction policy, which enables us to further optimize the action strategy A of the seller (i.e., the recommender system) by maximizing its utility (payoff) U_S [49]. In this section, we show that this can be formulated based on von Neumann-Morgenstern's *expected utility theory*. We then specify the formulation in terms of three example payoff objectives, i.e., click-through rate, sales revenue and consumption diversity.

4.4.2.1 Expected Utility Maximization

Because of the uncertainty/risk inherent in the game, it is nature to formulate action optimization as *decision making under uncertainty*. Consider a given game G_u between the seller S and a specific buyer u, the action space is the set of all possible combinations of l goods, $\mathcal{A} = \mathcal{I}^L$. An action $A \in \mathcal{A}$ yields an outcome $Z = (A, R) \in \mathcal{Z} = \mathcal{A} \times \mathcal{R}$ with probability distribution $p(Z)$ (*aka* lottery), where the reaction space $\mathcal{R} = A \cup \{\emptyset\}$. Because our knowledge about the environment is imperfect, we would rather adopt a probabilistic action strategy such that actions for G_u are sampled according to a distribution $p_u(A)$ defined over the action space $\mathcal{A}$ and any $A \in \mathcal{A}$ is taken with probability $p_u(A)$, then we have $p_u(Z) = p_u(A)p_u(R|A)$, where we specify the dependence on the user with a subscript to emphasize the fact that the action is customized for each user.

A utility function $U_S(Z)$ is a mapping $U_S : \mathcal{Z} \to \mathbb{R}$, which defines a preference relation $\succcurlyeq$ over the outcome space $\mathcal{Z}$ such that $Z \succcurlyeq Z'$ if and only if $U_S(Z) \geqslant U_S(Z')$. Without loss of generality, we assume $\succcurlyeq$ is von Neumann-Morgenstern rational, i.e., it satisfies the four axioms: completeness, transitivity, independence and continuity[2]. The von Neumann-Morgenstern (vNM) theorem defines the best outcome of a decision in an environment under uncertainty as follows[64].

THEOREM 2 [EXPECTED UTILITY]: *Suppose $\succcurlyeq$ is a preference defined by an utility function U_S that satisfies the 4 axioms, for any two distributions (lotteries) $p(Z)$ and $q(Z)$, we have: $p \succcurlyeq q$ if and only if $\mathbb{E}_p(U_S) \geqslant \mathbb{E}_q(U_S)$.*

Proof. c.f. [64]. □

Based on the vNM theorem, the optimal action strategy $p_u(A)$, given $p_u(R|A)$, can be achieved by the following linear optimization:

$$\max_{p_u(A)} \sum_{A\in\mathcal{A}} p_u(A) \sum_{R\in\mathcal{R}} p_u(R|A)U_S(A,R) \tag{15}$$
$$s.t. : \sum_{A\in\mathcal{A}} p_u(A) = 1, \text{ and } p_u(A) \geqslant 0.$$

A simplex solution for the above is given simply by:

$$p_u(A) = \delta_{A,A^*_u} \text{ where } A^*_u = \arg\max_A \sum_{R\in\mathcal{R}} p_u(R|A)U_S(A,R).$$

In practice, it is usually favorable, (e.g., for risk-robustness reasons) to choose a less sparse distribution (i.e., a portfolio [57]) rather than the singular distribution as defined by a simplex solution, the discussion of which is, however, beyond the scope of this work.

4.4.2.2 Action Strategy Parametrization

Although the simplex solution looks simple, exhaustive search throughout the outcome space is still something practically prohibitive as it requires $O(NM^{l+1})$ lookups. To this end, we propose to parameterize the action distribution in terms of a small set of parameters Θ, e.g., to assume action A is sampled from a parametric distribution $p_u(A;\Theta)$. In this way, we can search $\mathcal{A}$ efficiently by optimizing Θ instead. As a preliminary study, here we devise a simple parametrization by randomizing a utility-based ranking scheme with a scalar parameter α. Particularly, for any given user u, assume the top-ranked l items (i.e., items with highest payoffs) are denoted $\{i^*_1, \ldots, i^*_l\}$, we generate the action A as follows:

- $A = \emptyset$.
- For j from 1 to l do:
  - With probability $(1-\alpha)$ add i^*_j to A
  - With probability α add an random item to A

This way, action optimization in Eq(15) become a one-dimensional optimization, to which the solution can be obtained efficiently, e.g., via golden-section search.

A more flexible parametrization is to factorize $p(A)$ sequentially as $p(A) = p(i_1) \times p(i_2|i_1) \times \ldots \times p(i_l|i_1, i_2, \ldots, i_{l-1})$, with a few simplifications, we can search the action space by dynamic programming. We leave this for future research.

[2]It is easy to verify that the three example payoffs we explore in this chapter satisfy these four axioms.

4.4.2.3 Strategic Payoff Specification

So far, our discussion of action optimization is in terms of an abstract payoff function U_S. We now specify our formulation with three concrete strategic objectives.

Payoff #1: Click-Through Rate (CTR). Click-through rate or CTR is the ratio of responses (i.e., $R_t \neq \emptyset$) out of all the interactions. CTR is the most important measure of success for many real-world recommender systems because it crucially determines so many important factors ranging from traffic, revenue to user base. For example, it corresponds to the advertisement click rate in Google, the movie rental rate in Netflix, the order placement rate in Amazon, and the rate of friend connection in Facebook Friend-Finder. CTR can be formulated in the CCF framework as follows:

$$CTR = \mathbb{E}_u[\mathbb{E}_A[p_u(R \neq \emptyset|A)]] = \sum_{u \in \mathcal{U}} f_u \sum_{A \in \mathcal{A}} p_u(A) p_u(R \neq \emptyset|A)$$

where f_u is a measure of user loyalty (e.g., user u's visit frequency), and

$$p_u(R \neq \emptyset|A) = 1 - \frac{\exp(\theta_u)}{\exp(\theta_u) + \sum_{i \in A} \exp(\phi_u^\top \psi_i)}.$$

Payoff #2: Sales Revenue (SR). Another important measure of success is sales revenue or SR, which is the revenue that a recommender system receives from the transactions (interactions) with the users. SR is a weighted version of CTR, i.e., each click is assigned a weight of importance. Based on CCF, SR can be formulated via:

$$SR = \mathbb{E}_u[\mathbb{E}_A[\mathbb{E}_{i \in A}[c_i p_u(R = i|A)]]] = \sum_{u \in \mathcal{U}} f_u \sum_{A \in \mathcal{A}} p_u(A) \sum_{i \in A} c_i p_u(R = i|A)$$

where c_i denotes the price (weight) of an item i.

Payoff #3: Consumptions Diversity (CD). It is widely believed that recommender systems are the key contributor that turns the industry from what used to be a highly concentrated "blockbuster" [3] towards a highly diversified long-tail (niche) market [12, 79]. Recent research shows that this is, however, not entirely true — a recommender system, if designed improperly, could reinforce consumption concentrations [22]. In order not to turn our society to a echo chamber, it is important to encourage consumption diversity (CD), i.e., to ensure the consumptions of the whole population are not narrowly concentrated. Moreover, CD is also important to online firms to help them gain profit from long-tail market. CD can be formulated based on the CCF framework in terms of expected choice entropy:

$$CD = \mathbb{E}_u[\mathbb{E}_A[H_u(R|A)]] = -\sum_{u \in \mathcal{U}} f_u \sum_{A \in \mathcal{A}} p_u(A) \sum_{i \in A} p_u(R = i|A) \log p_u(R = i|A)$$

where $H_u(R|A) = \sum_{i \in A} p_u(R = i|A) \log p_u(R = i|A)$ is the entropy of user u's choice in the context of A.

[3] The well-known 80-20 rule or the Pareto principle states that, of the many goods available, consumptions are concentrated on a small subset of bestselling ones.

4.4.3 Implications of CCF and Future Work

We finally remark that there are some interesting properties of the proposed CCF model. Firstly, since the games in user-system interactions are finite, there exists an equilibrium point (i.e., a stable strategy). As a matter of fact, since that the reaction to a given action is rational and that the action given $p_u(R|A)$ is vNM-rational, it can be shown that the CCF model is sequentially rational and therefore achieves the *perfect Nash equilibrium* [49]. From a practical point of view, it is, however, possible to optimize the recommender systems more aggressively beyond the market equilibrium. Particularly, the analogy of recommender system to a monopoly market provides a number of important perspectives, e.g., the reflection of *price discrimination* in recommender system — how recommender system can exploit its market power to gain consumer surplus [12]. Another interesting topic is to explore the correlation and conflict of goods, and optimize action A as a bundle based on portfolio theory [57, 4]. We would rather leave these interesting discussions for future research.

4.5 Experiments

We test the proposed CCF model on a real-world commercial recommender system. Because CCF is comprised of two components, it is necessary to test each of them separately — otherwise, it would be difficult to tell if a change of performance is due to one component or the other or both. Our experiments therefore consist of two test-beds. Firstly, we compare CCF with CF baselines in terms of their abilities in preference estimation, which has already been done in Chapter 3. This comparison gives us an idea on how effective our model for $p_u(R|A)$ is compared with state-of-the art CF models. Furthermore, we compare the CCF framework (i.e., MLF + Action optimization) and the current recommendation scheme (i.e., CF + utility-based ranking). This comparison further demonstrates how the game-theoretic formulation, particularly how action optimization, further enhance the design of a recommender system. The later is what we report here.

We evaluate the entire CCF framework (i.e., MLF + action optimization) in terms of its ability to achieve the three strategic goals.

Evaluation metrics. We test a recommendation model by applying it on top of the algorithm in production and comparing the results with the production baseline. To assess performance, we report the relative surplus. In particular, let m denote one of the three measures (i.e., click-through rate, sales revenue and consumption diversity), a relative surplus score is defined by:

$$\text{relative surplus} = \frac{m(\text{model}) - m(\text{production})}{m(\text{production})}$$

Evaluation protocol. To illustrate how effective each of the CCF components is, we compare *CCF with action optimization* (CCF+AO), to *CCF without action optimization* (CCF-AO) as well as the conventional recommendation scheme (*collaborative filtering with utility-based ranking* or CF+RK). For each model, we simulate its relative surplus score by applying the model to the production output. In particular, we take the top 50K users who visit our website most frequently as test probes and trace them for one month. For each of these user u and each of the dates d, we maintain a positive set $P_{u,d}$ and a negative set $N_{u,d}$ by including all the articles that user u reads on date d into $P_{u,d}$ and any other items in the content pool of date d into $N_{u,d}$. We assume user u turns to take items only from $P_{u,d}$

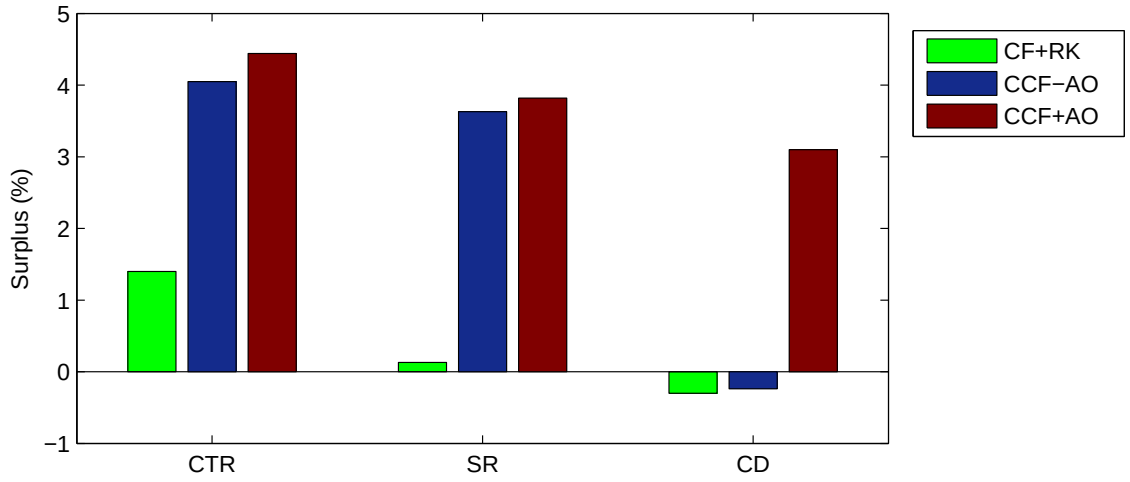


Figure 4: Performance in achieving strategic objectives: relative surplus compared to the production baseline in terms of CTR, SR and CD.

and ignores those in $N_{u,d}$ on date d. Specifically, the reaction of user u on date d to any action A is assumed as follows: for any item $i \in A$, if $A \cap P_{u,d} \neq \emptyset$ and $i \in A \cap P_{u,d}$, u takes i with probability $1/|A \cap N_{u,d}|$ or otherwise ignores it; a nonresponded session occurs when $P_{u,d} = \emptyset$. To compute sales revenue, we randomly assign to each item a positive number as "price", which is predefined and never changed throughout the evaluation. Moreover, maximizing consumption diversity alone leads to meaningless random recommendations; to this end, we impose a hard constraint to ensure that the decreases in CTR is no more than 0.5%.

Results and analysis. The aggregate results on the 50K probe users are depicted in Figure 4. Applying a traditional recommendation scheme (CF + preference based ranking) on top of the production baseline only yields marginal improvements in CTR and SR. In contrast, the CCF model gains up to 4.5% and 3.9% surplus in CTR and SR respectively; and action optimization further significantly enhance these numbers. In terms of consumption diversity, our experiment confirms the findings of [22] since applying CF and CCF directly without consideration of CD inevitably leads to consumption concentration, as shown by the negative surplus scores in Figure 4. In contrast, CCF+AO is the only one among the three models that yields positive surplus in CD. In particular, with less than 0.5% reduction of CTR, it gains up to 3.2% improvement of diversity. These observations are somewhat surprising considering that the preliminary action parametrization we used in the experiment is a bit overly-simplistic — it merely contains one single parameter α (c.f. Section 4.4.2.2). In future work, we plan to explore more flexible forms of action parametrization such as the sequential factorization model mentioned in 4.4.2.2; we expect to have even more promising results.

4.6 Summary

We presented a novel game-theoretic framework for recommendation by viewing the user-system interactions at recommender system as buyer-seller interactions in a monopoly economic market. Since the decisions of the user and the buyer are interdependent, this new

perspective motivates us to optimize the action strategy of the system by first predicting users' reaction and then adapting its action to maximize the expected payoff. The proposed CCF framework consists two essential components: (1) a model for $p_u(R|A)$ that integrates choice models in econometrics and latent factor model in collaborative filtering to encode the notion of collaborative games; and (2) a formulation for optimizing system action A in terms of expected strategic payoffs such as click-through rate, sales revenue and consumption diversity. Experiments on a real-world commercial recommender system have demonstrated the effectiveness and appealing promise of the proposed framework.

So far, we have investigated behavior prediction in recommender systems where users make their individual decisions independently. This is, however, not the case in systems with social networking functionalities (e.g., in online social networking systems), where users are interacting with one another and social contagion plays an significant role — the interactions dramatically influence individual behavior of decision making. Starting the next Chapter, we investigate behavior prediction in a social environment and develop models and algorithms to capture the interplay between social interactions and individual behavior.

CHAPTER V

MODELS FOR HOMOPHILY

The first part of the thesis (i.e., the previous two chapters) focuses on modeling individual behavior of decision making as in a recommender system without considering social interactions. In the second part which includes this and the next two chapters, we further examine behavior in social communities where users are actively interacting with one another and social contagion plays an significant role in influencing individual behavior due to fundamental social effects (e.g., Homophily, trust, influence), [93, 95, 19].

A social network is not solely a *social graph* that connect users with their friends, but it is also an *interest graph* that connect users with the resources (e.g., game, ad, brand) they like. One problem of fundamental interest is: how to effectively propagates both friendship and interest through the network. In this chapter, we show that the information contained in *interest networks* (i.e. user-service interactions) and *friendship networks* (i.e. user-user connections) is highly correlated and mutually helpful. We propose a framework [93] that exploits the social effect of "Homophily" to establish an integrated network linking a user to interested services and connecting different users with common interests, upon which both friendship and interests could be efficiently propagated. The proposed *friendship-interest propagation* (FIP) framework devises a factor-based random walk model to explain friendship connections, and simultaneously it uses a coupled latent factor model to uncover interest interactions. We discuss the flexibility of the framework in the choices of loss objectives and regularization penalties and benchmark different variants on the Yahoo! Pulse social networking system. Experiments demonstrate that by coupling friendship with interest, FIP achieves much higher performance on both *interest targeting* and *friendship prediction* than systems using only one source of information.

5.1 Introduction

Online social networking services have brought to the public a new style of social lives parallel to our day-to-day offline activities. Popular social network sites, such as Facebook, Linkedin and Twitter have already gathered billions of extensively acting users and are still attracting thousands of enthusiastic newbies each day. Doubtlessly, social networks have become one of today's major platforms for building *friendship* and sharing *interests*.

Fundamental to all social network services is the goal to effectively model the *interests* of a user and the *friendship* between users [69]. On the one hand, by capturing a user's interests and accordingly exploiting the opportunity to serve her/him with potentially interesting service items (e.g. news, games, advertisements, products), one can improve the satisfaction of a user's participation and boost the revenue of a social network site as well (e.g. via product purchases, virtual transactions, advertisement clicks). On the other hand, connecting people with common interests is not only important for improving existing users' loyalty, but also helps to attract new costumers to boost the site's traffic. In fact, *friendship prediction* (a.k.a. link prediction) and *interest targeting* (a.k.a. service recommendation) are two important tools available in almost all the major social network sites. Both activities which occur routinely in a social network have accrued a tremendous wealth of interaction

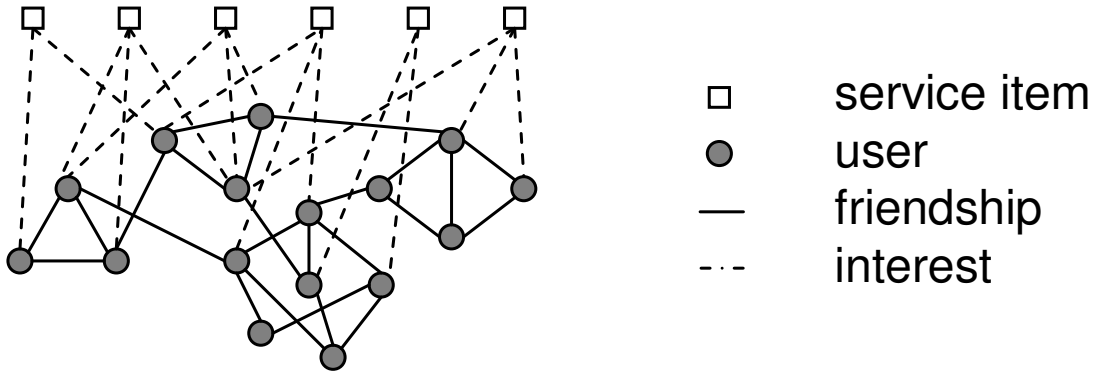


Figure 5: An example of *friendship network* and *interest network*.

traces, both *among users* (i.e. friendship network) and *between users and service items* (i.e. interest network). Figure 5 depicts a typical topology of a heterogeneous graph in the context of social networks.

5.1.1 Interests and Friendship

Modeling user interests and friendship in social networks raises unique challenges to both research and engineering communities. The information about a user's behaviors is often scattered in both friendship and interest networks, involving other users that are closely connected to the user and different activities that the user has engaged in. A fundamental mechanism that drives the dynamics of networks is the underlying social phenomenon of *homophily* [61]: people with similar interest tend to connect to each other and people of similar interest are more likely to be friends.

Traditional user profiling approaches often do not take full advantage of this fact. Instead they either employ feature engineering to generate hand-crafted meta-descriptors as fingerprint for a user [75, 23] or they extract a set of latent features by factorizing a user's registered profile data; for example, by means of sparse coding [45] or latent Dirichlet allocation [10]. These approaches could be inaccurate because neither user friendship nor user behavior information is taken into account.

Recent approaches resort to collaborative filtering (CF) techniques [16, 71, 1, 41] to profile user interests by collaboratively uncovering user behaviors, where users are assumed to be unrelated to each other. While CF performs well in recommendation systems where decisions are mainly made individually and independently, it could fail in the context of social networks where user interactions substantially influence decision making [37, 61].

Modeling friendship is equally challenging. A typical social network is a graph both large and sparse, involving hundreds of millions of users with each being connected to only a tiny proportion of the whole virtual world. This property rules out traditional spectral algorithms for graph mining [65, 66] and calls for algorithms that are both efficient to handle large scale connections and capable of reliably learning from rare, noisy and largely missing observations. Unfortunately, progress on this topic to date is limited [48].

5.1.2 Friendship Interest Propagation

This chapter exploits the important role homophily plays in social networks. We show that friendship and interest information is highly correlated (i.e. closely-connected friends tend to have similar interests) and mutually helpful (i.e. much higher performance for both friendship prediction and interest targeting could be achieved if coupling the two processes to exploit both sources of evidence simultaneously). We present a friendship-interest propagation (FIP) model that integrates the learning for interest targeting and friendship prediction into one single process.

The key idea in FIP is to associate latent factors with both users and items, and to define coupled models to encode both interest and friendship information. In particular, FIP defines a shared latent factor to assure dynamical interaction between friendship network and interest network during the learning process. In doing so, FIP integrates both interest and friendship networks to connect a user to both items of potential interest and other users with similar interests. FIP hereby provides a single unified framework to address both link prediction and interest targeting while enjoying the resources of both sources of evidence. Experiments on Yahoo! Pulse demonstrate that, by coupling friendship with interest, FIP achieves much higher performance on both tasks.

The contributions of this work are three-fold:

1. We present the friendship-interest propagation model that propagates two different types of evidence through heterogeneous connections.
2. We formulate the FIP model in a computational framework, discuss the flexibility in the choices of loss objectives (e.g. ℓ_2, logistic regression, Huber's loss) and regularization penalties (e.g. sparse coding, ℓ_2 penalties) and we benchmark different variants in a real-world social networking system;
3. For the implementation of FIP, we present a built-in scheme for bias correction based on pseudo-negative sampling to avoid overfitting, and we also deliver an optimization package that allows distributed optimization on streaming data.

5.2 Problem Definition

We begin by briefly reviewing the state-of-the-art. This will come in handy as we will link them to our model in §5.3.

Modeling dyadic interactions is the heart of many web applications, including link prediction and interest targeting. Typically, a pair of instances from two parties (such as users and items), $u \in \mathcal{U}$ and $i \in \mathcal{I}$, interact with each other with a response $y_{ui} \in \mathcal{Y}$. The mapping

$$\{(u, i) \to y_{ui} \text{ where } u \in \mathcal{U}, i \in \mathcal{I}\}$$

constitutes a large matrix $Y \in \mathcal{Y}^{|\mathcal{U}| \times |\mathcal{I}|}$, of which only a tiny proportion of entries are observable; the goal is to infer the value of a missing entry $y_{\tilde{u}\tilde{i}}$, given an incoming pair $(\tilde{u}, \tilde{i})$. Essentially, the observed interactions define a graph, either unipartite (when $\mathcal{U} = \mathcal{I}$) or bipartite. The task amounts to propagating the sparse observations to the remainder (unobserved) part of the matrix. For convenience we will henceforth reserve u, v as *user* index and i, j as *item* index.

5.2.1 Interest Targeting

Interest targeting, or (service) recommendation, works with a bipartite graph between two different parties, e.g. user u and item i. It aims at matching the best item i^* to a given

user u. We consider collaborative filtering (CF) approaches, which tackle the problem by learning from past interactions. Existing CF models can be generally divided into two categories: *neighborhood models* [72, 60] and *latent factor models* [41, 1] as we have reviewed in Chapter 2.1 and 3.3. It is also possible to combine the neighborhood models and latent factor models. A recent example is discussed [40], where the basic idea is to apply the *locality of dependencies* directly to the latent factors, for example:

$$\hat{\phi}_u = \frac{\sum_{v \in \Omega_u} \omega_{uv} \phi_v}{\sum_{v \in \Omega_u} \omega_{uv}} \quad y_{ui} \sim p(y_{uv} | \hat{\phi}_u^\top \psi_i; \Theta). \tag{16}$$

This model[1] which is quite similar to [40] was deployed on the Netflix data yielding significantly better performances over both pure-neighborhood and pure latent factor models.

5.2.2 Friendship Prediction

Friendship (link) prediction recommends users to other users in the hope of acquainting people who were previously not connected in the network (or even unfamiliar with each other). Unlike interest targeting, the user network is unipartite. For a pair of users (u, v) the observation whether they are connected is a binary value c_{uv}. Link prediction crucially influences both the traffic and the revenue of a social network and it is hence recognized as one of the key tasks in social network analysis.

Ideally, our goal is to learn a distribution over jointly exchangeable matrices (e.g. by applying the Aldous-Hoover factorization theorem). For reasons of practicality we pick a finite-dimensional factorization instead, which we shall discuss in the next section. Before we do so, let us briefly review existing approaches. Some of them employ random walk methods [50, 70] or spectral graph algorithms [65, 66].

Random Walk. As we have reviewed in Chapter 2.2, a random walk on the graph C is a reversible Markov chain on the vertexes $\mathcal{U}$. The transition probability from the vertex u to vertex v is defined $p(v|u) = c_{uv}/d_u$. Here d_u denotes the degree of vertex u; c_{uv} the connection weight between nodes u and v. Vertexes are considered close whenever the hitting time is small or whenever the diffusion probability is large.

Spectral Algorithms. For the given network C, the un-normalized Laplacian is defined by $L = D - C$, where D is a diagonal matrix with $D_{uu} = d_u$. Spectral algorithms diffuse the connections by maximizing the spectral smoothness to obtain the intrinsic kinship defined by the dominant eigenvectors of the Laplacian:

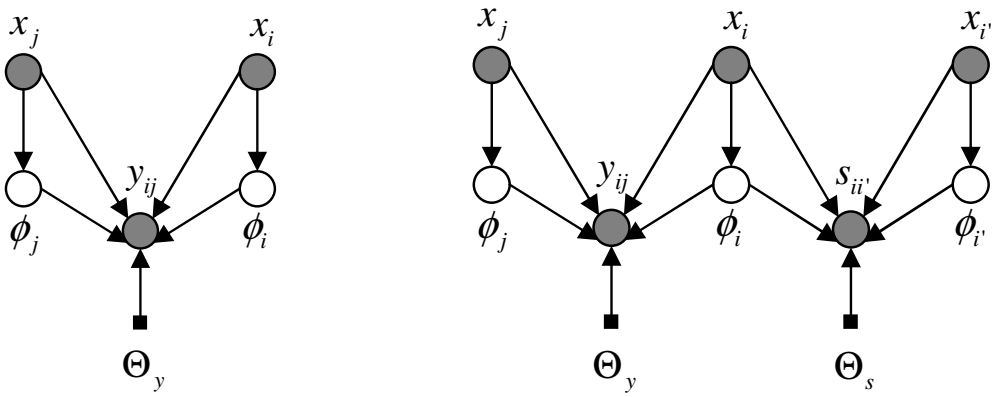
$$\sum_{u,v} c_{uv} \left\| z_u - z_v \right\|^2 = 2ZLZ^\top, \text{ where } Z = [z_1, \ldots, z_{|\mathcal{U}|}]. \tag{17}$$

5.3 Friendship-Interest Propagation

We now consider interest targeting and link prediction in the context of social network, where evidence for both *interest* and *friendship* are available, allowing us to solve both tasks in a single framework. The rationale is that *friendship* and *interest* information are to some degree correlated,[2] i.e. the network exhibits homophily [61] and the propagation of friendship and interest would be mutually reinforcing if modeled jointly.

[1] In this case the set of neighbors Ω_u contains u with $\omega_{uu} = 1$.

[2] Empirical analysis on Yahoo! Pulse illustrates that the interest correlation (Pearson score, max 1.0) between two directly-linked friends is 0.43, much higher than average.



(a) (b)

Figure 6: Graphical representations of regression based latent factor model (RLFM) and friendship-interest propagation (FIP) model.

The nontrivial correlation between interest and friendship motivates joint modeling of both sources of evidence. As shown in Figure 6, the friendship-interest propagation(FIP) model simultaneously encodes the two heterogeneous types of dyadic relationships: the user-item interactions $\{y_{ui}|u \in \mathcal{U}, i \in \mathcal{I}\}$, and user-user connections $\{c_{uv}|u, v \in \mathcal{U}\}$. Our model is built on latent factor models.

5.3.1 Modeling interest

To characterize the user-item dyads, y_{ui}, we assume that for each user u and item i there exist observable properties x_u (e.g. a user's self-crafted registration files) and x_i (e.g. a textual description of a service item)[3]. Moreover, we also assume that there exist some subtle properties which cannot be observed directly, such as a user's interests, a service item's semantic topics. We denote these latent features by ϕ_u for u and ψ_i for i respectively. We assume the response y_{ui} depends on both types of features (i.e. observable and latent):

$$\phi_u \sim p(\phi_u|x_u) \quad \psi_i \sim p(\psi_i|x_i) \quad y_{ui} \sim p(y_{ui}|\phi_u, \psi_i, x_u, x_i, \Theta),$$

where Θ denotes the set of hyper-parameters. To design a concrete model, one needs to specify distributions for the dependencies, $\phi_u|x_u$, $\psi_i|x_i$, and $y_{ui}|x_u, x_i, \phi_u, \psi_i$.

This model is essentially an integration of collaborative filtering [1] and content filtering [17]. On the one hand, if the user u or item i has no or merely non-informative observable features such that we have access to only their identity and past interactions, the model degrades to a factorization-style collaborative filtering algorithms [71]. On the other hand, if we assume that ϕ_u and ψ_i are irrelevant, for instance, if u or i is totally new to the system such that there is no interaction involving either of them as in a cold-start setting [104], this model becomes the classical feature-based recommendation algorithms [16, 102, 17], which predict the interaction response y_{ui} purely based on the observed properties of u and i, and are commonly used in, e.g. webpage ranking [102], advertisement targeting [16], and content recommendation [17].

[3]Whenever we do *not* have access to these properties we simply default to the expected value of the latent variables, which is easily achieved in a probabilistic model.

5.3.2 Modeling friendship

We now extend the interest model to incorporate the social friendship-connection information among users. For this purpose, we define a random walk process for user-user networking. But unlike traditional random walk models [50, 70], we assume a user u is fully characterized by her observable features x_u and latent factor ϕ_u, and devise the following model for user-user transition:

$$\phi_u \sim p(\phi_u|x_u, \Theta) \text{ and } c_{uv} \sim p(c_{uv}|\phi_u, \phi_v, x_u, x_v, \Theta), \tag{18}$$

where c_{uv} reflects an observed state transition from u to v. Unlike in random walk models where proximity in a graph is simply used to smooth secondary estimators of parameters (e.g. reachability, hitting times), we make direct use of it to model the latent variables ϕ_u. Note that whenever we restrict the norm of ϕ_u (e.g. by ℓ_2 regularization) and when we use an inner product model $\phi_u^\top \phi_v$ to assess similarity, we approximately recover the graph Laplacian of Eq(17).

In this way our model integrates two different methodologies — collaborative filtering and random walks. It is different from traditional random walk models in which transition probability is defined solely based on graph topologies. It is also different from traditional CF models in that it is defined on unipartite dyadic relationships. By doing so, this integrated model not only allows learning of latent factors to capture graph topologies, but it also alleviates certain critical issues in random walks: for example, it naturally handles heterogeneous graphs (e.g. a compound graph consisting of both unipartite and bipartite connections such as Figure 5), and it also makes applicable computationally-efficient sequential learning algorithms (e.g. stochastic gradient descent), avoiding directly manipulating large matrices.

5.3.3 The joint model

Based on the above descriptions, we finally summarize the overall FIP model in Figure 6 and the table below. Note that the tuples (u, x_u, ϕ_u) now play "double duty" in encoding interest interactions (u, i, y_{ui}) and friendship connections (u, v, c_{uv}) simultaneously. Learning shared factors from coupled relationships gives us both more evidence and more constraints to work with, and in turn leads to better generalization.

The Friendship-Interest Propagation (FIP) model.	
$\forall\ u \in \mathcal{U}$	$\phi_u \sim p(\phi_u\|x_u, \Theta)$
$\forall\ i \in \mathcal{I}$	$\psi_i \sim p(\psi_i\|x_i, \Theta)$
$\forall\ u \in \mathcal{U}, i \in \mathcal{I}$	$y_{ui} \sim p(y_{ui}\|\phi_u, \psi_i, x_u, x_i, \Theta)$
$\forall\ u, v \in \mathcal{U}$	$c_{uv} \sim p(c_{uv}\|\phi_u, \phi_v, x_u, x_v, \Theta)$

5.4 The computational framework

5.4.1 Model specification

So far we deliberately described the FIP model in terms of general dependencies between random variables to make it explicit that the model is quite a bit more general than what can

be achieved by an inner product model. Here, we specify the model within an optimization framework.

For computational convenience we assume linear dependencies between x_u and ϕ_u plus a noise term[4] ϵ. This means

$$\phi_u = Ax_u + \epsilon_u \text{ where } \mathbb{E}[\epsilon_u] = 0. \tag{19}$$
$$\psi_i = Bx_i + \epsilon_i \text{ where } \mathbb{E}[\epsilon_i] = 0. \tag{20}$$

ϵ is typically assumed to be Gaussian or Laplace. Whenever nonlinearity in x is desired we can achieve this simply by using a feature map of x and an associated kernel expansion. Finally, we assume that the dyadic response (e.g. y_{ui}) depends on latent features only through the inner product (e.g. $\phi_u^\top \psi_i$) and on observable features through a bilinear product (e.g. $x_u^\top W x_i$) [17]. That is:

$$y_{ui} \sim p(y_{ui}|f_{ui}) \text{ where } f_{ui} = \phi_u^\top \psi_i + x_u^\top W x_i.$$
$$c_{uv} \sim p(c_{uv}|h_{uv}) \text{ where } h_{uv} = \phi_u^\top \phi_v + x_u^\top M x_v.$$

Here, assume $x_u \in \mathbb{R}^m$ and $x_i \in \mathbb{R}^n$, the matrices $W \in \mathbb{R}^{m \times n}$ and $M \in \mathbb{R}^{m \times m}$ provide a bilinear form which captures the affinity between the observed features for the corresponding dyads. We also impose Laplace or Gaussian priors on W and M. One advantage of using an ℓ_1 (i.e. Laplace) prior is that it introduces sparsity, which makes Eq(19) equivalent to sparse-coding [45] and thus improves both compactness and predictiveness of the learned latent factors ϕ and ψ.

Given observed responses for the dyads $\{(u,i) \in \mathcal{O}_y\}$ and $\{(u,v) \in \mathcal{O}_c\}$, the problem of minimizing the negative log-posterior of FIP boils down to the following objective:

$$\begin{aligned} \min \lambda_y \sum_{(u,i)\in\mathcal{O}_y} \ell(y_{ui}, f_{ui}) + \lambda_c \sum_{(u,v)\in\mathcal{O}_c} \ell(c_{uv}, h_{uv}) \\ + \lambda_{\mathcal{U}} \sum_{u\in\mathcal{U}} \gamma(\phi_u|x_u) + \lambda_{\mathcal{I}} \sum_{i\in\mathcal{I}} \gamma(\psi_i|x_i) \\ + \lambda_W \Omega[W] + \lambda_M \Omega[M] + \lambda_A \Omega[A] + \lambda_B \Omega[B], \end{aligned} \tag{21}$$

where $\lambda_\cdot$'s are trade-off parameters, $\ell(\cdot,\cdot)$ denotes a loss function for dyadic responses. The term $\gamma(\phi|x) = \Omega[\phi] + \gamma_x(x,\phi)$. Here $\Omega[\cdot]$ is used to penalize the complexity (i.e. ℓ_2, ℓ_1 norm). The term $\gamma_x(x,\phi)$ regularizes ϕ by fitting the observed feature x, as defined by Eq(21). This type of regularization are equivalent to applying content factorization (e.g. LSI, NMF, LDA) to the feature x in terms of a factor ϕ and bases A^{-1} or B^{-1}.

The motivations for a computational framework instead of direct probabilistic inference are mainly two-fold: First, the two formulations are somewhat equivalent — the distribution of the dyadic response (e.g. y_{ui}) and its dependence on the prediction (e.g. $p(y_{ui}|f_{ui})$) can be encoded precisely through the choice of loss functions; likewise, the prior over the observations or parameters could also be readily translated into the regularization penalties. Secondly, computational models allow more scalable algorithms, e.g. via stochastic gradient descent, whereas probabilistic reasoning often requires Monte Carlo sampling or quite nontrivial variational approximations.

[4]Note that the latent 'noise' term is actually meaningful. It indicates the deviation of the user/item profiles from its cold-start estimates Ax_u and Bx_i respectively.

5.4.2 Loss

In our case, both y and c are binary, i.e. $y_{ui}, c_{uv} \in \{\pm 1\}$. We performed an extensive study in our experiments comparing a large variety of different loss functions. For the convenience of optimization, we limit ourselves to differentiable (in many cases, also convex) loss functions (see also Figure 7 for details):

Least Mean Squares: This is the most popularly-used loss in matrix factorization. It minimizes the Frobenius norm of the prediction residue matrix and leads to a SVD-style algorithm. We have the loss

$$\ell_2(y, f) = \frac{1}{2}(1 - yf)^2. \tag{22}$$

Lazy Least Mean Squares: This is a slight modification of ℓ_2 loss for the purpose of classification [92]. Basically, it is an iteratively truncated version of the ℓ_2 loss via

$$ll_2(y, f) = \min(1, \max(0, 1 - yf)^2). \tag{23}$$

It has been shown that this loss approximates the classification error rate in the example space [92].

Logistic regression: This is the loss used in a binary exponential families model. It is given by

$$log(y, f) = \log[1 + \exp(-yf)]. \tag{24}$$

Huber loss: This is the one-sided variant of Huber's robust loss function. It is convex and continuously differentiable via

$$\eta(y, f) = \begin{cases} \frac{1}{2}\max(0, 1 - yf)^2, & \text{if } yf > 0. \\ \frac{1}{2} - yf, & \text{otherwise.} \end{cases} \tag{25}$$

Ψ loss: Unlike other loss functions, which are all convex upper bound of the 0-1 loss, the Ψ loss [74] is non-convex. Both theoretical and empirical studies have shown appealing advantages of using non-convex loss over convex ones, such as higher generalization accuracy, better scalability, faster convergence to the Bayes limit [74, 92]. We implement the following version:

$$\Psi(y, f) = \begin{cases} \frac{1}{2}\max(0, 1 - yf)^2, & \text{if } yf > 0. \\ 1 - \frac{1}{2}\max(0, 1 + yf)^2, & \text{otherwise.} \end{cases} \tag{26}$$

5.5 Optimization and Implementation

5.5.1 Optimization

Minimizing Eq(21) is a nonconvex problem regardless of the choice of the loss functions and regularizers due to its use of bilinear terms. While there *are* convex reformulations for some settings, they tend to be computationally inefficient for large scale problems — the convex formulations require the manipulation of a full matrix which is impractical for anything beyond thousands of users. Moreover, the relationships between users change over time and it is desirable to have algorithms which process this information incrementally. This

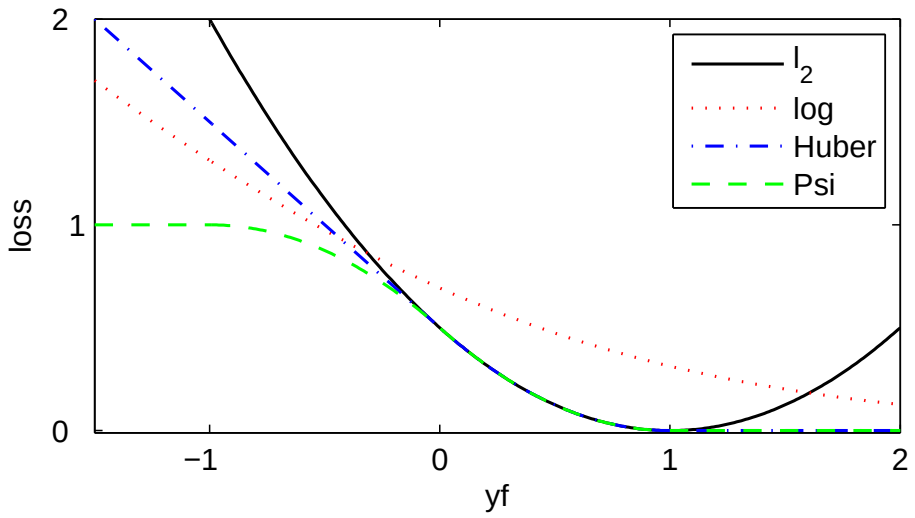


Figure 7: Least mean squares (ℓ_2), logistic (log), Huber and Ψ-loss (Psi) loss functions for binary classification.

calls for learning algorithms that are efficient and amendable to dynamic updating so as to reflect upcoming data streams, rendering less attractive those offline learning algorithms such as classical SVD-based CF algorithms or spectral link prediction methods that involve manipulation of large-scale matrices. This requirement becomes more important for FIP than for traditional latent factor models because we are now dealing with two (instead of one) large-scale coupled interactions and feature observations.

We established algorithms for distributed optimization based on the Hadoop MapReduce framework. The basic idea is to decompose the objective in Eq(21) by optimizing with respect to y_{ui} and c_{uv} independently in the Map phase, and to combine the results for ϕ_u in the Reduce phase.

5.5.2 Bias Correction

A key challenge for learning latent factors from dyadic interactions is that the observations are extremely sparse with almost exclusively positive interactions observable. That is, we typically do *not* observe explicit information that user u does *not* like item i. Rather, the fact that we have not observed (u, i) suggests that u might not even know about i. In other words, absence of a preference statement or a social link should not be interpreted *absolutely* as negative information.

At the same time, unless we have access to negative signals, we will almost inevitably obtain an estimator that is overly optimistic with regard to preferences (e.g. predict positive for all the interactions). To balance both requirements we draw uniformly from the set of unobserved tuples (u, i) and (u, v) respectively and we require that, on average, observed pairs are preferred to unobserved ones.

In practice, since we use a stochastic gradient algorithm for minimization, for every positive observation, e.g. $y_{ui} = 1$, we randomly sample a handful set of missing (unobserved) entries $\{y_{uj}\}_{j=1:m}$, and treat them as negative examples (e.g. $y_{uj} = -1$,) with credibility $1/m$ each. Since the sampling procedure is random, the set of pseudo-negatives changes at each iteration and consequently each missing entry is treated as a potentially *very weak*

negative instance.

5.6 Discussion

This section provide a brief discussion on how our model is related to the models discussed in §5.2.

Our first observation is that our FIP model indirectly induces a kernel for the friendship network graph: $\mathsf{k}(u,v) = \phi_u^\top \phi_v$ via the learned embedding ϕs . This is similar to the information diffusion kernel for graph [39, 44] in that both kernels inherent a Riemannian manifold for $\mathcal{U}$ defined by the friendship network C, rather than a flat Euclidean space as in traditional CF models (e.g. neighborhood [72], factorization [71], RLFM [1]). However, it is also worth mentioning that the FIP induced kernel is different from diffusion kernels in that (i) our feature mapping ϕ_u is obtained from latent-factor based random walk model rather than topology-based random walk; and (ii) our model defines a compact low-rank manifold rather than a manifold that is potentially of infinite dimensionality [39, 44].

Traditional latent factor CF models [1, 71] work in Euclidean space where user factors ϕ_u are assumed identically and independently distributed: $\phi_u \sim \mathcal{N}(0, \sigma^2 I)$. Our model relates ϕ_u with one another by modeling the social network graph. This is equivalent to a row-correlated matrix-Gaussian $\Phi \sim \mathcal{MN}(0, \Sigma \otimes I)$, where $\Phi = [\phi_1, \ldots, \phi_{|\mathcal{U}|}]^\top$, $\mathcal{MN}$ is a matrix-variate Gaussian, and $\Sigma \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{U}|}$ defines the row-covariance (user-user covariance). By inexplicably modeling the friendship manifold, our model hereby generalizes traditional latent factor CF models from Euclidean space to Riemannian space, in a way analogous to how diffusion kernels generalize Gaussian kernels.

Note that, although the neighborhood based latent factor model [40] also induces a manifold structure for $\mathcal{U}$, this manifold is virtual as it is directly constructed from Euclidean representations that essentially reflect the same amount of information. Our model generalizes this model by exploiting the true connections from social networks.

The FIP also differs from traditional link prediction algorithms. Actually, it borrows the idea of latent factor CF models to model the transition probability in terms of latent factors, making possible that (i) latent factors can be learned from network topologies; and (ii) connections can be propagated collaboratively through the interaction between latent factors. Essentially, our approach establishes an integrated network of interest and friendship that connects people with similar interests, and upon which both friendship and interests could be efficiently propagated.

5.7 Experiments

We demonstrate the FIP model on Yahoo! Pulse in terms of both interest targeting and friendship prediction.

5.7.1 Yahoo! Pulse Data

Yahoo! Pulse (`pulse.yahoo.com`) is a social network site that allows users to create profiles, connect to friends, post updates, and respond to questions, as in other social networks. More importantly, it provides a mechanism for users to share interests, i.e. users can upload, download, install applications, and invite friends to try interesting applications. Our motivation is to utilize the user-user friendship network and user-application interest network from Yahoo! Pulse, so as to simultaneously propagate interest and friendship. We

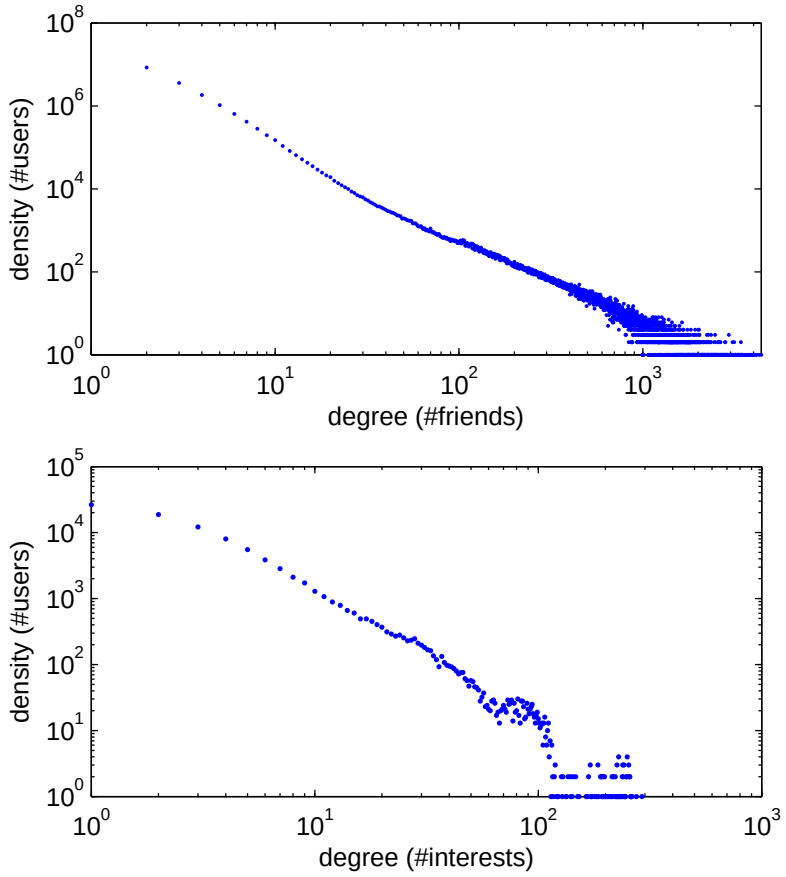


Figure 8: Degree distributions of the Yahoo! Pulse friendship and interest networks.

examine data collected on Yahoo! Pulse for about one year, involving hundreds of millions of users and a large collection of applications, such as games, sports, news feeds, finance, entertainment, travel, shopping, and local information services. Figure 8 shows the degree distribution of this data set. The data is very sparse and almost half of the users only have one friend connections and do not like any of the applications (they are essentially not using the network). Our goal is to propagate evidence to establish reliable connections both among users and between users and applications.

We use a subset of Yahoo! pulse data. The data set has 386 application items, 1.2M users, 6.1M friend connections and 29M interest interactions. There is a significant difference in the densities of the two networks in this data set. As the item set is pretty small, the interest network is relatively dense – each user likes 23.5 items on average. In contrast, as the user population is large, the friendship network is extremely sparse: on average, each user only has 4.9 friends out of the total 1.2 million.

5.7.2 Evaluation Metrics

Both interest targeting and link prediction lead to a ranking of entities (e.g. items and users that the system may recommend) according to a score function. In our context this means that the scores f_{ui} and h_{uv} induce a ranking. Hence it is natural to use ranking metrics to assess performance. We consider the following three scores:

Table 6: Comparison of service recommendation performance.

Models	loss	$\Omega[\cdot]$	AP@5	AR@5	nDCG@5
SIM			0.630	0.186	0.698
RLFM			0.729	0.211	0.737
NLFM			0.748	0.222	0.761
FIP	ℓ_2	ℓ_2	0.768	0.228	0.774
FIP	lazy ℓ_2	ℓ_2	0.781	0.232	0.790
FIP	logistic	ℓ_2	0.781	0.232	0.793
FIP	Huber	ℓ_2	0.781	0.232	0.794
FIP	Ψ	ℓ_2	0.777	0.231	0.771
FIP	ℓ_2	ℓ_1	0.778	0.231	0.787
FIP	lazy ℓ_2	ℓ_1	0.780	0.231	0.791
FIP	logistic	ℓ_1	0.779	0.231	0.792
FIP	Huber	ℓ_1	**0.786**	**0.233**	**0.797**
FIP	Ψ	ℓ_1	0.765	0.215	0.772

AP is the *average precision*. AP@n averages the precision of the top-n ranked list of each query.

AR is the *average recall* of the top-n rank list of each query.

nDCG or *normalized Discounted Cumulative Gain* is the normalized position-discounted precision score. It gives larger credit to top-ranked entities.

In all three metrics we use $n = 5$ since most social networks and recommendation sites use a similar number of items for friend and application suggestions; it is also the standard recommendations size used in the current system. For our evaluation we use cross-validation, where we randomly partition the data into two equally sized pieces and use one for training and the other for testing. All three measures are computed on testing data only, and they are averaged over five random repeats.

5.7.3 Interest Targeting

In this section, we report the results on interest targeting (i.e. application recommendation). We adopt a fairly strict evaluation by assessing the top results out of a total preference ordering of the item set for each user. In particular, for each user u, we consider all the 386 items as candidates; we evaluate the recommendation performance by assessing the quality of the top-5 items based on the comparison between ground truth (the actual list of the applications that user i liked) and the top-5 ranked shortlist outputted by each model.

For comparison, we take three popular CF models as baseline: the item-oriented neighborhood model (SIM), the regression based latent factor model (RLFM) [1], and the combination of them (referred to as neighborhood based latent factor model or NLFM [71]). SIM and RLFM use interest information; NLFM use both friendship information and interest information.

We test the baselines and different variants of FIP model, each of which is referred to in terms of the name combination of a loss and a regularization (e.g. FIP(ℓ_2, ℓ_2)). Table 6 demonstrate the overall results, i.e. the mean value of metrics averaged over 5 random runs. As the scale of the data is quite large, the predictive variance is very small (less than 0.002) and it is therefore not reported.

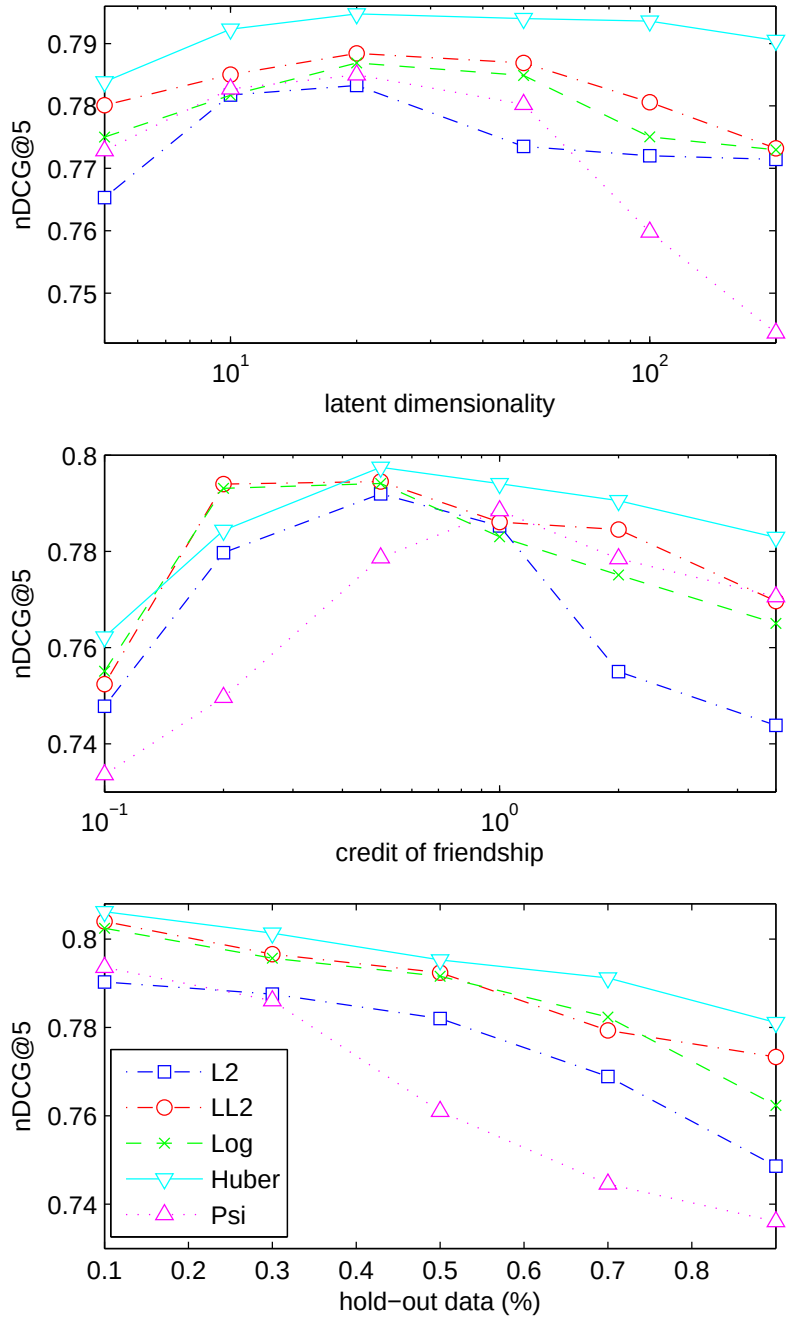


Figure 9: Service recommendation performance as a function of latent dimensionality, friendship credibility and the proportion of hold-out data.

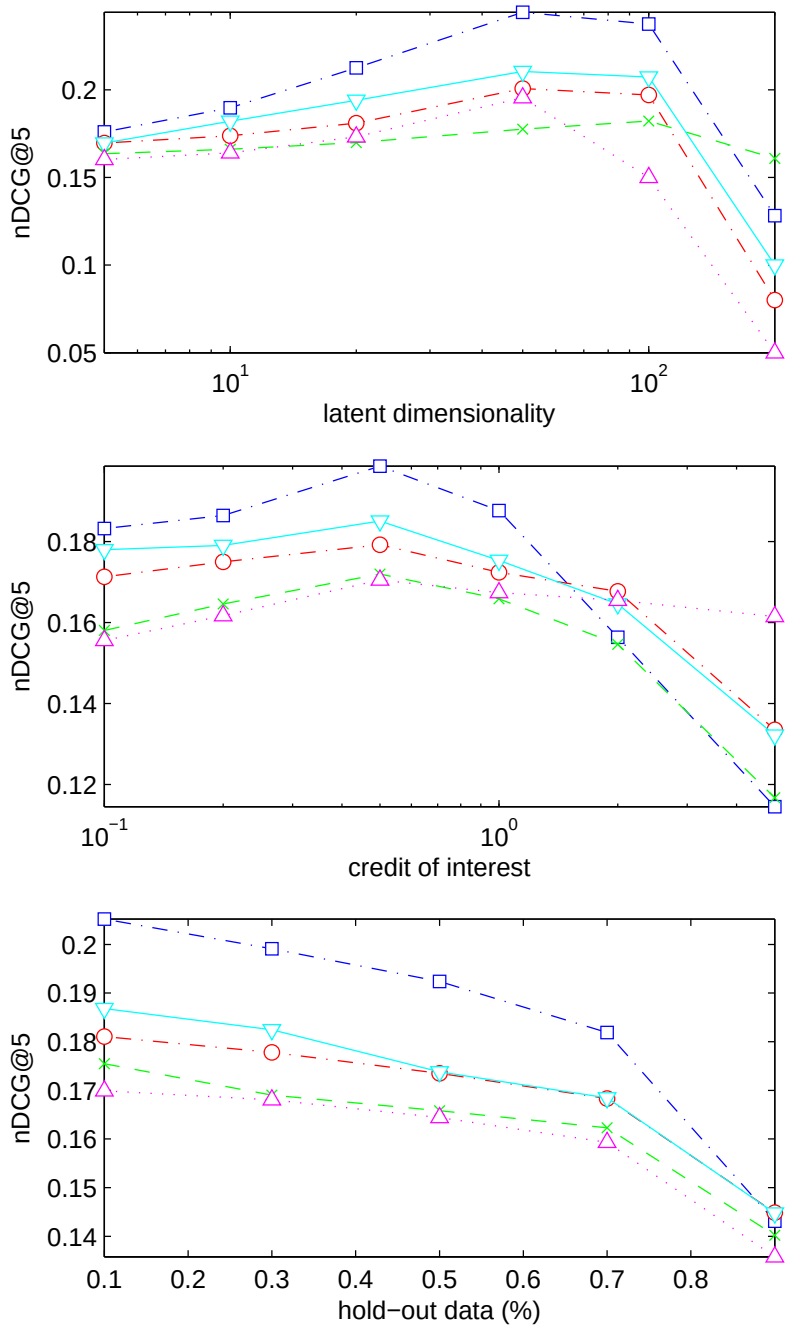


Figure 10: Friend prediction performance as a function of latent dimensionality, interest credibility and the proportion of hold-out data.

For the relatively dense interest network, all the reported models in our system actually achieve satisfactory performance in interest propagation. For most models, both the nDCG@5 and AP@5 scores are above 0.7, that is, out of the five recommended items, on average 3.5 are truly "relevant" (i.e. actually being liked by the user). Such performance is sufficiently satisfactory for propagating the 386 approved services in the current interest network. With such good performance, there is really not much room for further improvement. However, we still observe that noticeable improvements are obtained by the FIP models. Specifically, in terms of the nDCG@5 scores (similar comparisons apply to other metrics), FIP outperform the SIM model by up to 11.4%, RFLM by up to 10.8%, and NLFM by up to 4.7%. All improvements are significant (according to t-test with confidence threshold 0.01).

Among the 5 loss options for FIP, the one-sided Huber loss, the lazy ℓ_2 loss and logistic regression perform equally well (with Huber slightly better). Surprisingly, the nonconvex Ψ loss performs very poorly, even worse than ℓ_2. We attribute this to the non-convexity of the Ψ loss – while nonconvex losses perform superiorly in learning linear classifiers [74, 92], they could totally fail in learning the bilinear form of latent factor models because of the strong nonconvexity.

With respect to the two types of regularization, we observe that for each loss the ℓ_1 regularizer almost consistently outperforms ℓ_2. As ℓ_1 regularization leads to compact (i.e. sparse) latent factors by assigning submissive latent dimensions to exactly 0, this observation suggests that sparseness can improve the informativeness of latent factors (being sparser implies smaller description length) and in turn leads to superior performance.

One of our claims is that friendship information is helpful for interest targeting. An interesting test would be to check how the credibility (i.e. λ_c) of the friendship influences the performance of interest prediction. We report this result in Figure 9. We can see that, as we increase λ_c, the performance first increases (it peaks at 0.5, or 1.0 for Ψ) and thereafter it starts to drop. This observation coincides with our intuitions: friendship information is truly useful for interest propagation, it helps interest targeting with discounted credit; yet, if too much weight is given to friendship, the latter may pollute the interest evidence and in turn harm interest targeting performance.

We also test the effects of two parameters: the dimensionality of latent factors k, and the proportion of hold-out testing data. Results are reported in Figure 9. For most losses, between 10 and 20 latent factors are sufficient for prediction. Also, with the exception of the Ψ loss, the performance is quite stable to both parameters. This observation validates our hypothesis of the Ψ loss not being very amenable to efficient optimization: as latent dimension increases, more local optima are created and the Ψ loss performs worse; likewise, as training data becomes more sparse, the Ψ loss may be trapped in worse local optima.

An important procedure in our implementation is the bias-correction, i.e. generating pseudo-negative samples to correct the selection bias, as described in §5.5.2. We demonstrate the effects by comparing results obtained with and without this procedure in Figure 11. The comparisons are striking, indicating that latent factor models, if trained without negative examples, turn to overfit the training observations and misleadingly predict "positive" for most dyads. Our algorithms, by sampling missing interactions and using them as very weak negatives, guide the latent factor to capture the dyadic interactions while avoiding being fooled by the positive-only observations.

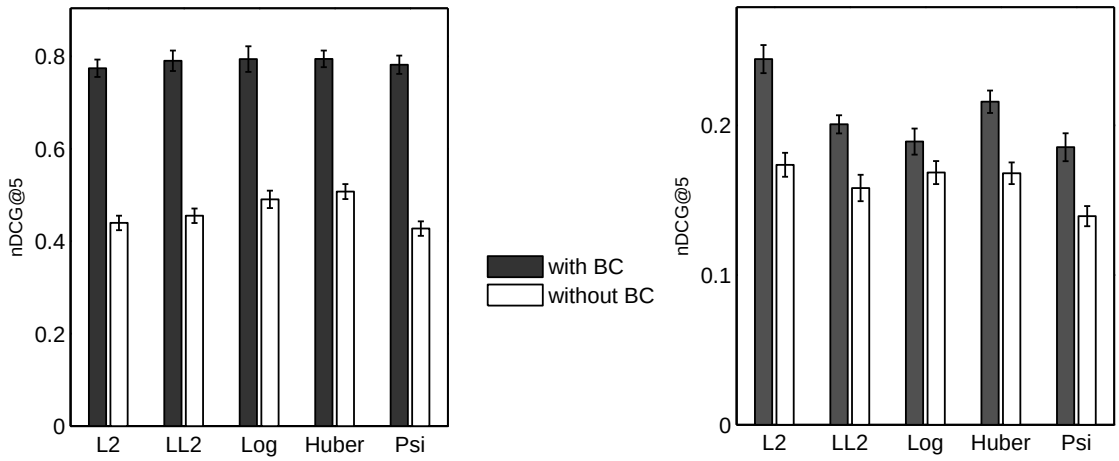


Figure 11: Recommendation performance with vs without bias-correction (BC): service recommendation (left) and friendship prediction (right).

Table 7: Comparison of friendship prediction performance.

Models	loss	$\Omega[\cdot]$	AP@5	AR@5	nDCG@5
RLFM			0.164	0.202	0.174
FIP	ℓ_2	ℓ_2	**0.359**	**0.284**	**0.244**
FIP	lazy ℓ_2	ℓ_2	0.193	0.269	0.200
FIP	logistic	ℓ_2	0.174	0.220	0.189
FIP	Huber	ℓ_2	0.210	0.234	0.215
FIP	Ψ	ℓ_2	0.187	0.255	0.185
FIP	ℓ_2	ℓ_1	0.186	0.230	0.214
FIP	lazy ℓ_2	ℓ_1	0.180	0.223	0.194
FIP	logistic	ℓ_1	0.183	0.217	0.189
FIP	Huber	ℓ_1	0.188	0.222	0.200
FIP	Ψ	ℓ_1	0.178	0.208	0.179

5.7.4 Friendship Prediction

We conducted similar evaluations on friendship propagation (i.e. link prediction). As the user population in the Pulse social network is huge, it is prohibitive to take all the users as candidate friends and generate a total ordering of the whole user set for each user to evaluate the prediction performance; similarly, the models relying on neighborhood information (e.g. SIM and NLFM) are no longer tractable as they require computations quadratic in the number of users. To this end, we use a different evaluation mechanism: for each user u, we randomly sample M users that are not connected to u, we mix them with the set of users that u actually connects to. We then use this probe-polluted set as candidates, upon which the ranking performance is computed. In our experiments, we use $M = 300$ random probes per user.

We report the overall results in Table 7, where RLFM is used as the baseline model. The friend-network is extremely sparse (0.0039% density). Propagating friendship based on such sparse evidence is much more difficult, for example, RLFM only achieves 17% AP@5

and nDCG@5, which means out of the top-5 recommendations, less than 1 is truly relevant. Yet, we observe a significant improvement, as high as a 40% gain in nDCG@5, when FIP is used. This observation indicates that there is strong evidence of homophily in the Pulse social network such that users with similar interests are truly interested in each other. By leveraging the relatively dense interest evidence to assist the extremely sparse friendship graph, FIP achieves much higher performance in friendship predictions.

Regarding the loss functions, this time the ℓ_2 loss performs the best. We hypothesize that for much sparser friendship networks, losses that are more suitable for classification tasks tend to overfit the observed connections by making prematurely *hard* decisions to exclude connections that are not observed at the training stage. Similarly, because the ℓ_1 regularizer makes the latent factor sparser (some components of ϕ are shrunk to be exactly 0), it also turns to make *hard* predictions and in turn performs worse. Indeed, we observe significantly better performance for the ℓ_2 loss and/or regularizer, which turn to make smoother decisions.

The effects of parameter settings on this task are reported in Figure 10. We observe similar trends as in the previous task, although the performance is more sensitive: the performance changes faster as latent dimensionality increases or training data decreases. This matches the bias-variance analysis of statistic learning: as friendship connections are extremely sparse, we are typically dealing with a small-sample-size estimation, for which decrease of training data (e.g. increase hold-out proportion) or increase of model complexity (e.g. increase latent dimensionality) will inevitably lead to the increase of either bias or variance or both, and therefore the models are more likely to overfit the training observations and in turn generalize poorly.

As before, Figure 11 (right) shows that bias correction significantly improves the performance. Note that the difference is not as large as in interest targeting. This is likely due to the observation sparsity: in the sparser friendship network, two users that were not observed in the training set still have a good chance to be friends, which means many pseudo-negatives could be false-negatives.

5.8 Related Works

Collaborative filtering (CF) and link prediction were previously studied separately in two different research communities. The proposed FIP model bridges these two methodologies with a unified model. Essentially, FIP embeds all the users and items into the same space (e.g. Euclidean, simplex) so that the distances between two entities (e.g. user-item, user-user) reflect the *relatedness* (e.g. interest, friendship) between them, and hence, provides a unified treatment for both interest targeting and link prediction.

Existing approaches to link prediction diffuse the sparse connections using topology-based random walk [50, 70, 103] or spectral graph algorithms [65, 66], both of which involve expensive manipulation of large matrices. FIP borrows the idea of latent factor models in collaborative filtering [71, 1, 41] and it shows connections to random walk based models. As a side effect we obtain computationally attractive algorithms for efficient random walks.

Traditional CF techniques exploit past records of user behavior for future prediction based on either *neighborhood based* or *latent factor based* methods. The neighborhood latent factor model [40] merged these two models and reported significant performance improvement on the Netflix data. Though promising, the network structure exploited in this combined model [40] is a virtual one, constructed using the same evidence for learning latent factors. Our FIP model extends this model to allow the actual social network structure to

be captured in latent factor learning.

Along another line, the recently proposed *regression based latent factor model* (RLFM) [1] incorporates node (user or item) features to improve recommendation performance in the *cold-start* scenario. The FIP model also generalizes RLFM [40] in a way analogous to how information diffusion kernels [39, 44] generalize the Gaussian kernels. Basically, instead of working in the Euclidean space as RLFM does, FIP induces a limited-dimensional Riemannian manifold defined by the topologies of both the unipartite friendship network and the bipartite user-service interaction network.

The FIP model has a close connection with recent works on collective matrix factorization [52, 84, 76, 103], where the tasks of learning relational data were also formulated in terms of factorizing multiple matrices. The current work continues our prior investigations on this topic and further examines interest and friendship propagation in the context of social networks, a task urgently motivated by emerging demands from social network services [69]. The techniques developed in this work also advances the state-of-the art from several aspects: (i) besides the dyadic relational data (i.e. edges), we also attempt to leverage the rich information conveyed by the node features using regression model similar to RLFM [1] or factorization models similar to sparse coding [45]; in this way, FIP integrates latent factor models [41, 71, 1] and predictive bilinear models [102, 17]; (ii) we present distributed optimization algorithms, address bias correction, discuss and benchmark different loss objectives and regularizers; and our work provides one of the first large-scale examinations of interest-friendship propagation in a real social network system.

One work relevant to ours is the social recommendation approach proposed by [55], where the trust relationships among users are used to improve cold-start recommendation. This model can be seen as a special case of our FIP model by assuming (i) no node feature x_i or x_j; (ii) ℓ_2 loss objective; and (iii) asymmetric factor based random walk model, i.e. the transition probability is modeled as a multiplicative function of user-factor and a basis. Also, this work did not address the task of user relationships (e.g. trust, friendship) propagation. Along this line, our work addresses both tasks with a more general framework and conducts large-scale evaluation on a social networking system.

5.9 Summary

In this chapter, we have shown the significance of Homophily in social behavior prediction — the behaviors (i.e., interest) of socially connected users are mutually reinforcing such that coupling the behavioral and social evidences improves behavior prediction. Nevertheless, as we show in the next chapter, the social contagion is way more complicated than the simple Homophily as the underlying assumption is largely debatable — socially connected users do not necessarily agree with one another when making decisions. Instead, a much stronger signal is the trust-distrust relationship, e.g., we are usually more willing to follow the opinion of someone we trust than that of someone we merely know on Facebook.

CHAPTER VI

MODELS FOR SIGNED SOCIAL TIES

In this Chapter, we examine a more complicated social contagion effect beyond the simple unipolar Homophily [95]. We study the problem of *labeling the edges* of a social network graph (e.g., acquaintance connections in Facebook) as either *positive* (i.e., trust, true friendship) or *negative* (i.e., distrust, possible frenemy) relations. Such signed relations provide much stronger signal in tying the behavior of online users than the unipolar Homophily effect, yet are largely unavailable as most social graphs only contain unsigned edges.

We show the surprising fact that it is possible to infer signed social ties with good accuracy *solely* based on users' behavior of decision making (or using only a small fraction of supervision information) via unsupervised and semi-supervised algorithms. This work [95] hereby makes it possible to turn an unsigned *acquaintance network* (e.g. Facebook, Myspace) into a signed *trust-distrust network* (e.g. Epinion, Slashdot). Our results are based on a framework that simultaneously captures users' behavior, social interactions as well as the interplay between the two. The framework includes a series of latent factor models and it accommodates the principles of balance and status from Social psychology. Experiments on Epinion and Yahoo! Pulse networks illustrate that (1) signed social ties can be predicted with high-accuracy even in fully unsupervised settings, and (2) the predicted signed ties are significantly more useful for social behavior prediction than simple Homophily.

6.1 Introduction

Social networks are playing an increasingly important role in shaping the business models of today's internet industry. Not only are most emerging online services providing functionality to facilitate social interactions in their systems (e.g. last.fm, slashdot, Flikr, Yelp), but even many traditional internet systems are actively exploiting social networks to enhance their services, marketing and revenue, e.g., search engines (e.g., Bing), online retailers (e.g., Amazon), Web portals (e.g., Yahoo!). This is partly due to the widespread faith in the social effect of *Homophily* [61] that people socially acquainted tend to behave similarly. However, the extent to which this belief holds true is debatable — after all, it is unreasonable to expect two users who know each other (e.g., who are connected in Facebook) to behave alike in every aspect of their online decision making (e.g., purchase, ad click, movie rental, search intent, political view, sentimental moods). Instead, a much stronger signal of social ties is the network of trust relationships, since in many decisions we make, the opinions from the people we *trust* are what we really care about and what our decisions usually get influenced by. Unfortunately, this information is largely unavailable given the fact that the vast majority of social networks only contain acquaintance relationships, without distinguishing whom you trust from whom you don't.

Quite surprisingly, we show it is possible to infer a signed label for an acquaintance relationship — positive as trust (true friendship), negative as distrust (possible frenemy). In this way we make possible to turn an unsigned *acquaintance network* (e.g. Facebook, Myspace) to an signed *trust-distrust network* (e.g. Slashdot, Epinion). Moreover, by allowing us to tie the behavior of an online user to *whom he trusts* rather than *whom he knows*,

the predicted signed relations are significantly more informative than Homophily for the purpose of behavior prediction.

6.1.1 Signed Social Ties

The main focus of this chapter is to predict the *signed social ties* (i.e., trust/distrust) out of the unsigned (i.e., acquaintance) relationships in social networks. In particular, given that two users are socially connected, we would like to infer whether the connection is positive (e.g., trust, true friendship) or negative (e.g., distrust, possible frenemy). Effective modeling of signed social ties is not only of practical value to many internet services but also of general interest to the social science community. In particular, by inferring the signed relations and tie the behavior of online users, we may improve social targeting to promote online services [67, 58, 54, 93]. For example, by providing better matched search results, News articles, games, advertisements, or products, one can improve user satisfaction and also boost the revenue of a website (e.g. via product purchases, virtual transactions, advertisement clicks). Moreover, models for signed social ties also have the potential to improve our understanding of social relations and our society, and in turn shed light to sociological principles in general [28, 68, 25, 47].

More importantly, by predicting signed social ties, we enable the social connections in a social network to reflect much richer relationships that include both *positive* (e.g. trust, approval, true friendship) and *negative* (e.g. distrust, opposition, frenemy) interactions. This is nontrivial as we substantially extend the acquaintance relationship in standard social networks (e.g. Facebook), and also the unipolar tie-strength analysis of traditional link prediction [66, 68, 25] to a meaningful new dimension since we are able to *en-sign*[1] an unsigned network by turning it to a signed network.

A number of recent papers [28, 13, 43, 46] started investigating signed relationships in social networks. These pioneering works, however, focus on the topology of *signed networks* (e.g. Epinion, Slashdot) where the signed connections are explicitly observed. Moreover, they seem to neglect a rather important source of information. In particular, while social tie is naturally reflected in and directly revealed by user's behavior (e.g. user u is most likely to trust user v if a large number of their opinions agree), such vital clues were not captured previously. As a result, the tools developed there can only be applied to signed networks. To our knowledge, we present the first set of tools which allow us to infer signed relations even in *unsigned social networks* (e.g. Facebook, Twitter, Myspace) where the signs of social relations are not observed but could be inferred from behavioral clues.

6.1.2 Behavior-Relation Interplay

The interplay between social relations and user's behavior of decision making is a key driving force underlying social activities [67, 58, 54]. On the one hand, social interactions can influence individual's behavior of decision making [37, 3] — for instance, users can learn from each other, following their trusted friends' good decisions and avoiding their mistakes. On the other hand, a user's behavior could in turn impact his relationships with others — for instance, a user with malicious behavior turns to get bad reputations from others. In this chapter, we present a framework for predicting the signed social ties by capturing

[1]With slight abuse of expression, here by "en-sign" we mean "to assign a sign to" something that is unsigned.

such interplay. Specifically, we explore the following "mixture of effects" assumptions about social relations and users' behavior of decision making:

1. The decision of a user is not decided solely by his own taste but also influenced by the opinions of other people whom he trusts.
2. The social relationship between two users depends not only on their intrinsic kinship but is also evolving according to the agreement between their opinions.

It is desirable to distinguish these two groups of factors when modeling behavior and social relationships. We present the *Behavior Relation Interplay* (BRI) model, a latent factor model that leverages behavioral evidence to infer social interactions and at the same time exploits the learned relations to tie users' behavior. The key idea in BRI is to associate latent factors with both users and items (i.e. decision options such as movies, products, applications), and to define coupled models to simultaneously characterize both the social interactions and behavioral evidences.

A key question is whether it is possible at all to infer signed ties in the absence of any labels. One of the more surprising results of our work is that it is, indeed, possible. To achieve this we design both unsupervised and semi-supervised algorithms for signed social tie prediction. In the unsupervised setting, we only observe the acquaintance connections but none of the sign labels, as is the typical case in the vast majority of the acquaintance social network sites such as Facebook. We show that BRI achieves fairly accurate predictions. The key idea is a mechanism called "*automatic sign determination*" that discovers the signs of social ties automatically from user activity data. We also examine semi-supervised prediction, where a small fraction of the sign labels are given. We show that BRI model provides comparable performance to the leave-one-out prediction results (i.e. all but one signs are observable) of the topology-based algorithm [46].

To further improve the prediction accuracy and deepen our understanding of signed social ties, we extend the BRI model to encode general sociological principles, particularly the theories of *status* and *structure balance* [28, 47]. One nice property of our model is that it naturally provides a mapping from a user to his social status and an interesting visualization of the network. We show the sociological principles are useful for signed social tie prediction.

Finally, we show that by predicting signed social ties, BRI models achieve significantly better performance in behavior prediction than models equipped with Homophily. In particular, BRI gains 5.8% improvement of RMSE in rating prediction on Epinion data, and 6.9% improvement of nDCG in top-k ranking on Yahoo! Pulse data.

6.2 Background

6.2.1 Problem Definition

We consider a typical scenario in social networks where users routinely make decisions while actively interacting with one another. This motivates us to analyze the interplay between an individual user's behavior of decision making and the tie between users. Suppose we are given a set of users

$$\mathcal{U} = \{1, 2, \ldots, N\}$$

and a set of items (e.g., News articles, advertisement, retailing products, movies)

$$\mathcal{I} = \{1, 2, \ldots, M\}.$$

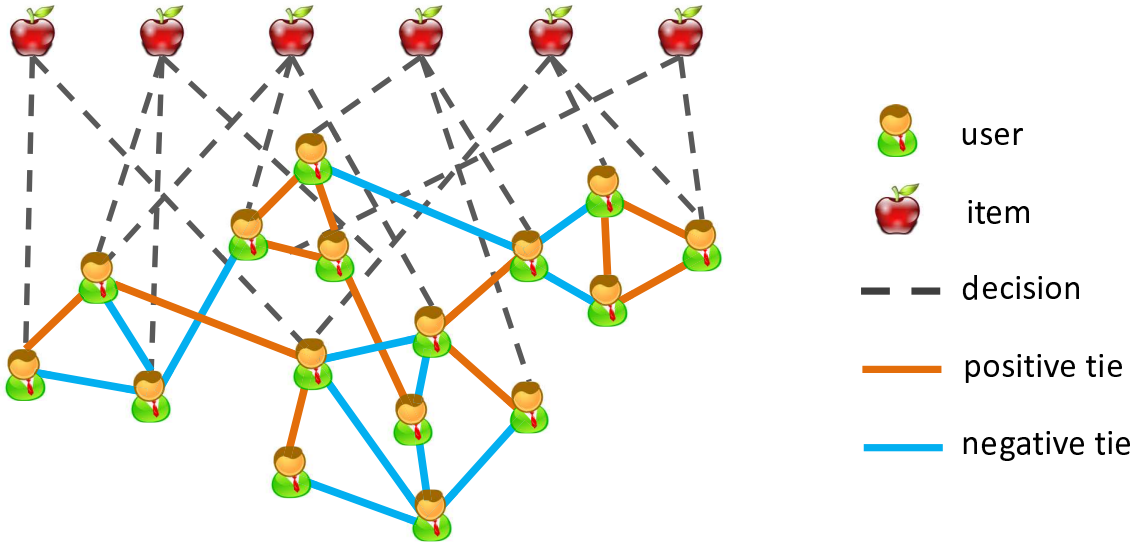


Figure 12: An illustration of decision making in social networks: users make decisions with respect to items while actively interacting with one another.

The users are connected by a social network represented by the graph $\mathcal{G}(\mathcal{U}, \mathcal{C})$. Here $\mathcal{C} = \{c_{uv}\}$ denotes the set of edges: $c_{uv} \in \{1, \text{missing}\}$ that define the connection between a pair of users $(u, v) \in \mathcal{U}^2$. In the network, users actively make decisions, for example, by clicking links, purchasing products, rating movies. Formally, a decision is a mapping

$$y : \mathcal{U} \times \mathcal{I} \to \mathcal{Y}.$$

That is: user u makes a decision regarding item i with a response $y_{ui} \in \mathcal{Y}$ (e.g. u rates movie i with a score of y_{ui}). For convenience, hereafter we use u and v to index users, and i and j for items unless stated otherwise.

We are motivated by the following two tasks:

En-signing social relations. A signed social tie is a mapping $s : \mathcal{C} \subset \mathcal{U}^2 \to \{\pm 1\}$ which characterizes the overall impression (trust/distrust) of user u to user v. In essence, our aim is to *label the edges* of the graph $\mathcal{G}$ (i.e. the acquaintance relationships in a standard social network) *with signed labels* $+1$ or -1.

Behavior prediction. We are also interested in predicting the users' behavior of decision making. In particular, given a user-item pair (u, i), we want to predict the response y_{ui} with high precision.

We focus on the more challenging task of sign labeling in §6.3 and §6.4, and defer the second goal to §6.5.

6.2.2 Related Work

To provide a context of our work, here we briefly review existing research on behavior prediction and relation analysis.

Behavior Prediction. Behavior prediction aims at predicting future (unseen) decision responses y_{ui} of a user u w.r.t an item i. We consider collaborative filtering (CF) approaches

[41, 1], which tackle the problem by learning from past behavior. A widely used approach to CF learns informative compact latent factors to uncover the dyadic interactions. The basic idea is to associate latent factors with each user and each item, and let the decision be explained by the interactions (e.g., multiplication) of these terms.

While CF approaches gain a lot success in recommender systems, they may be insufficient in the context of social networks where user interactions influence decision making dramatically [37]. For example, instead of making decisions purely by following his own taste as in recommender systems, in a social network, a user can achieve more productive decisions by learning from his friend's experiences. This aspect has been rapidly increasing in importance as many online services are starting to add a social network aspect to their system (e.g. last.fm, Yelp, Flikr) or directly rent existing social networks (e.g. Amazon, Netflix, Bing).

The previous chapter exploits Homophily for behavior prediction, where the behavior of two socially-connected users is tied and reinforced by each other. This chapter further investigates a more realistic case where acquainted users are not necessarily reinforcing each other in decision making. We do so by modeling the signed social ties, a much stronger signal that allows us to tie the behavior of a user to whom he trusts rather than whom he knows.

Unsigned Relation Analysis. Traditional relation analysis [50, 70, 66, 25] focuses on unsigned relations and estimates the tie strength between two users by propagating the observed links through the network topology. For a pair of users (u, v) the observation whether they are connected (i.e., know each other) is binary $s_{uv} \in \{1, \text{missing}\}$. Many existing methods employ random walk or spectral algorithms. A random walk on the graph $\mathcal{G}$ is a reversible Markov chain on the vertices $\mathcal{U}$. The transition probability $u \to v$, expressed via $p(v|u)$ is defined e.g. by $p(v|u) = d_u^{-1} s_{uv}$, where d_u denotes the degree of vertex u. Vertices are considered close whenever the hitting time is small or whenever the diffusion probability is large. In some cases spectral smoothness is used to obtain similarity via the associated graph Laplacian [78]. This yields the intrinsic kinship defined by the dominant eigenvectors of the graph [50].

Signed Relation Analysis. Social relations are complex and subtle [47, 28], reflecting interactions that could be either *positive* (e.g. trust, agreement, approval) or *negative* (e.g. distrust, disagreement, opposition). Yet until recently a number of authors began to investigate signed relations by examining the links in emerging signed social networks (e.g., Slashdot, Epinion). [28] applied random walk algorithms to propagate both positive and negative links on Epinion network. [43] conducted similar experiments on Slashdot data. [47] examined the applicability of social psychology theories at multiple signed networks. [46] further applied classification approaches to predict the sign of social relationships. While the goal of [46] is seemingly similar to the sign-labeling task we explore here, this work is still within the scope of link analysis since the focus there is solely on the network topology. Furthermore, they adopted a leave-one-out setting — a rather optimistic scenario where the sign labels for all but one edge are observable. Instead, our focus is not on network topology, but rather on discovering the interplay between relations and behavior. Also, we require no or fewer labels to obtain comparable prediction as [46].

6.2.3 Data

Before providing a formal description of our approaches let us briefly discuss the data, which we will use to draw intuitions for model development, test proposed models and motivate

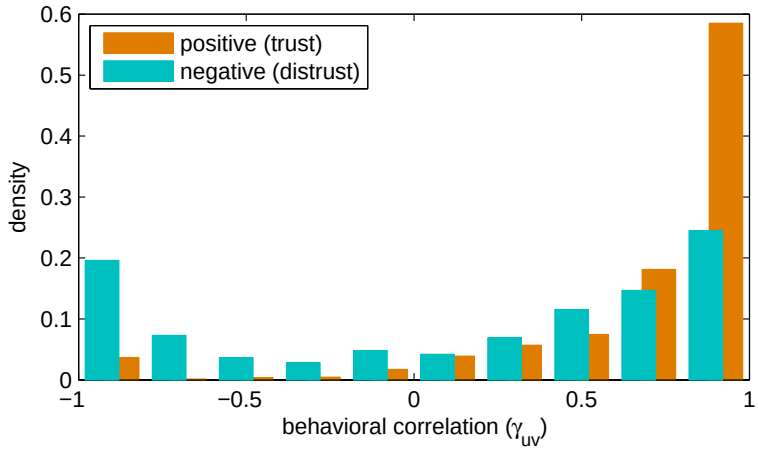


Figure 13: Histograms of behavioral correlations for positively and negatively connected users on `epinion.com`.

further enhancements.

Epinion We use data from `www.epinion.com` for our analysis as it is the only publicly available dataset we are aware of that contains both signed social relationship *and* user behavioral data. Epinion is a well-known knowledge and opinion sharing web site, where users post reviews and assign ratings (on a scale from 1 to 5) to various of items such as retailing products, companies, or movies. More interestingly, the site maintains a signed social network that allows each user to indicate explicitly other users that he is connected to as either positive (trust) or negative (distrust). We apply our models to predict the trust/distrust relationships. The data set contains 132 thousand users, 1560 thousand items, 13.6 million ratings and 850 thousand signed relationships. Both the user-item matrix and the user-user network are very sparse, with densities of only 0.014% and 0.0048% respectively.

Yahoo! Pulse `pulse.yahoo.com` is an unsigned social network that allows users to communicate with friends and also express their preference for items with explicit indications of "like". We examine data collected over one year, involving 10^8 users and a large collection of items, such as games, sports, News feeds, finance, travel, shopping, and local information services. Our evaluation focuses on a random subset consisting of about 400 items, 1.2 million users and 29 million "like" indications. Due to the unsigned characteristic of this network, we are not able to use this data set to evaluate edge-sign prediction. Instead, we will use this data set for behavior prediction evaluation (Section 6.5).

6.2.4 Motivating Observations

This chapter attempts to predict signed social ties by capturing the interplay between behavioral and social interactions. A key insight is that we consider the behavioral data (e.g. the ratings) to be useful for predicting signed social interactions. This is reasonable as in reality we are often willing to follow the opinions from a trusted friend and fight against those from a foe. As a result, decisions of positively-connected (i.e., trust) friends are more likely to agree, whereas for frenemies the chance of disagreement would be considerably higher. Here we empirically validate this intuition on the Epinion data, as can be seen in

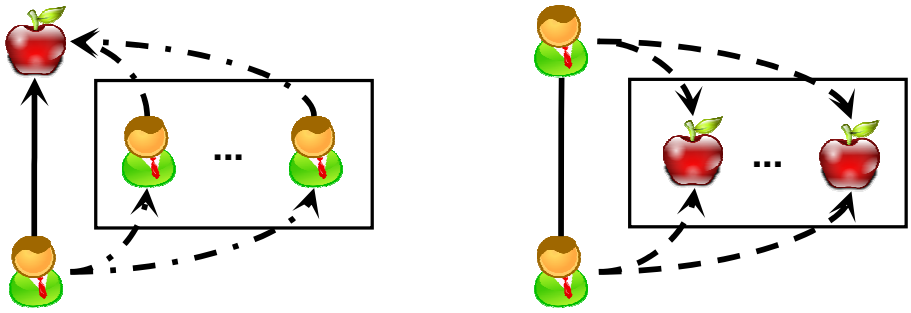


Figure 14: An illustration of the "mixture of effects" assumption in individual behavior and social relation.

Figure 13.

In particular, we quantify behavioral agreement using the Pearson correlation score between the ratings of two users, and we compare this between users with positive and negative links. The Pearson score is defined as follows:

$$\gamma_{uv} := \frac{\sum_{i \in \mathcal{O}[uv]} y_{ui} y_{vi}}{\sqrt{\sum_{i \in \mathcal{O}[uv]} y_{ui}^2 \sum_{i \in \mathcal{O}[uv]} y_{vi}^2}}, \tag{27}$$

where $\mathcal{O}[uv]$ denote the set of items that both u and v rate. Figure 13 plots the distribution of γ_{uv} for positive-relationships (u trusts v) and negative-relationships (u distrusts v). We can see that the behavioral correlation is highly predictive for signed relations. Specifically, the majority (almost 90%) of the positive links indeed show positive behavioral correlations. The case for negative links is a bit more complicated — the distribution peaks at both -1 and +1, indicating that users with distrust relationships could agree or disagree with each other. But even in this case, we can still observe that for almost 40% of the negative edges, the signs of relations are consistent with the signs of behavioral correlations.

6.3 Behavior Relation Interplay

The nontrivial relationship between behavioral correlation and the sign of social relations motivates us to leverage both social and behavioral evidences when predicting signed social ties (and/or behavior vice versa). In this section, we present models to capture the interplay between users' behavior of decision making and social interaction in the context of social networks.

6.3.1 The BRI Model

Similar to the latent factor CF models for behavior prediction, we devise a latent factor model for user behavior data $\{y_{ui}\}$. We associate latent factors ϕ_u with user u and ψ_i with item i. For notation convenience we require that each latent factor contains a constant component so as to absorb user/item-specific offset into latent factors. In particular, we assume the following model for behavior (The gist of our approach is captured by Figure 14):

Intrinsic taste We assume user u's interest in item i is quantified by $\phi_u^\top \psi_i$.
Trust credit Similarly, we assume user u trusts an acquainted user v with a credit $\phi_u^\top \phi_v$.

Social decision making Furthermore, we assume that user u makes decision w.r.t. item i by taking a random walk, that is: with probability p he follows his own taste $\phi_u^\top \psi_i$, or with probability $(1-p)$ he asks one of his friends, say v (with transition probability $\phi_u^\top \phi_v$) for advice, where v's opinion is captured by $\phi_v^\top \psi_i$. As a results, aspects that are favored/disliked by his friends should impact an individual user's own choices. The decision y_{ui} therefore depends on $p\phi_u^\top \psi_i + (1-p)\sum_v (\phi_u^\top \phi_v)\phi_v \psi_i$, which is achieved equivalently by "coloring" the inner product between users by a normalized variant of the correlation

$$M_y := (\operatorname{tr} C_y)^{-1} C_y \text{ where } C_y = \sum_v \phi_v \phi_v^\top \tag{28}$$

$$f_{ui} = \phi_u^\top \left[pI + (1-p)M_y\right] \psi_i \tag{29}$$

Here $p \in [0,1]$ encodes the the probability with which a user's behavior is decided by his own taste.

The same model can be applied to incorporate social relations among users. We assume a user u is fully characterized by its latent factor ϕ_u and the following model to characterize the relationships of user u to user v:

Prior impression We assume user u trusts an acquainted user v a priori (i.e., prior to any decision making behavior) with credit $\phi_u^\top \phi_v$.

Behavior agreement User u calibrates his impression regarding v based on the agreement between his own and v's decisions, e.g., according to the correlation $\sum_i (\phi_u^\top \psi_i)(\phi_v^\top \psi_i)$.

Social tie We assume the social interaction is a mixture of both prior impression and behavior agreement with mixture parameter q, i.e.: the social tie depends on $q\phi_u^\top \phi_v + (1-q)\sum_i (\phi_u^\top \psi_i)(\phi_v^\top \psi_i)$, or equivalently:

$$M_s := (\operatorname{tr} C_s)^{-1} C_s \text{ where } C_s = \sum_i \psi_i \psi_i^\top \tag{30}$$

$$h_{uv} = \phi_u^\top \left[qI + (1-q)M_s\right] \phi_v \tag{31}$$

Here $q \in [0,1]$ defines the mixture probability of prior impression. Unlike in random walk models where proximity in a graph is simply used to smooth secondary estimators of parameters (e.g. reachability, hitting times), we make direct use of it to model the latent variables ϕs.

Based on the above descriptions, we specify the probabilities with spherical Gaussian distributions (extensions to other distributions are straightforward) and we summarize the overall model in the table below.

The Behavior Relation Interplay (BRI) model.	
$\forall\ u \in \mathcal{U}$	$\phi_u \sim \mathcal{N}(\phi_u \vert 0, \sigma_{\mathcal{U}}^2 I)$
$\forall\ i \in \mathcal{I}$	$\psi_i \sim \mathcal{N}(\psi_i \vert 0, \sigma_{\mathcal{I}}^2 I)$
$\forall\ u \in \mathcal{U},\ i \in \mathcal{I}$	$y_{ui} \sim \mathcal{N}(y_{ui} \vert f_{ui}, \sigma_{\mathcal{Y}}^2 I)$
$\forall\ u, v \in \mathcal{U}$	$s_{uv} \sim \mathcal{N}(s_{uv} \vert h_{uv}, \sigma_{\mathcal{S}}^2 I)$

Here $\sigma_{\mathcal{I}}$, $\sigma_{\mathcal{U}}$, $\sigma_{\mathcal{S}}$, $\sigma_{\mathcal{Y}}$ are scalars specifying the variance.

We develop inference algorithms for the proposed model, relying on three types of dyadic evidences:

1. The user behavior trace $\{y_{ui}\}$ in the form of user-item interactions such as a user's rating for a movie.
2. The unsigned user-user social connections (e.g., the acquaintance relations at Facebook) $\{c_{uv}\}$, where $c_{uv} = 1$ when u and v know each other or missing otherwise.

   Or alternatively whenever available:
3. The signed connections $\{s_{uv}\}$, $s_{uv} = \pm 1$ or missing, e.g., the trust/distrust or friend/foe relations at Epinion and Slashdot.

We refer to c_{uv} as *unlabeled tie* and to s_{uv} as *labeled tie*. Our goal is to infer the labels of social ties using none (i.e. unsupervised) or only a fraction (i.e. semi-supervised) of the label observations by fitting the latent factor model BRI on the above three sources of evidences.

6.3.2 Unsupervised BRI

We first consider the case where no label is observed. This is a typical setting for the majority of social network sites such as Facebook. Our key idea is to use an absolute-value link function $|\xi|$, which bridges the *hidden* signed-ties we aim to infer, h_{uv}, to the unsigned acquaintance connections we actually observed, c_{uv}. In this way, we enable the mechanism of *automatic sign determination* in the same spirit as the maximum margin clustering of [87] — we assume the objective function achieves optimality at either $h_{uv} = +1$ or $h_{uv} = -1$; through optimization, the model will automatically determine the optimal sign for h_{uv} because different signs correspond to different objective values.

Formulation. Given the observations of user-item behavioral interactions $(u, i) \in \mathcal{O}_y$ and unsigned user-user connections $(u, v) \in \mathcal{O}_c$, we have the following optimization:

$$\underset{\phi,\psi}{\text{minimize}} \ \lambda_y \sum_{(u,i)\in\mathcal{O}_y} (y_{ui} - f_{ui})^2 + \lambda_c \sum_{(u,v)\in\mathcal{O}_c} (c_{uv} - |h_{uv}|)^2 + \lambda_{\mathcal{U}} \sum_{u\in\mathcal{U}} ||\phi_u||^2 + \lambda_{\mathcal{I}} \sum_{i\in\mathcal{I}} ||\psi_i||^2 \quad (32)$$

where $\lambda_{\mathcal{U}}$, $\lambda_{\mathcal{I}}$, λ_c, λ_y are trade-off parameters, f_{ui} and h_{uv} are defined according to Eq(29) and Eq(31).

Prediction. Given an incoming pair of users, (u, v), we predict the sign of the social tie (i.e., trust or distrust) as follows:

$$\hat{s}_{uv} = \text{sign}(h_{uv} - h_0) \quad (33)$$

Here h_0 is a threshold which suitably determines the fraction of positive and negative ties. Whenever we have access to some labels on a validation set, it is straightforward to determine h_0 via line search: simply sort h_{uv} by magnitude and compute the balanced error for each distinct value of h_{uv}. This is in complete analogy to the line search procedure of [35] for F_β scores.

Optimization. Eq(32) is minimized by stochastic gradient descent. Strictly speaking the objective does not decouple entirely in terms of ϕ and ψ due to the matrices M_y and M_s. Nonetheless we found that an approximation which performs stochastic gradient descent on ϕ and ψ while keeping M_y, M_s fixed produces good results. Also, it can be decomposed into sessions and parallelized using the MapReduce framework.

The stochastic gradient descent procedure is entirely standard: the algorithm processes all observations and updates the parameters in the direction of the negative gradient. For

instance, for $\mathcal{O}_y$ the updates are as follows:

$$\phi_u \leftarrow (1 - \eta\bar{\lambda}_{\mathcal{U}})\phi_u - \eta(f_{ui} - y_{ui})(p\psi_i + (1-p)M_y\psi_i) \quad (34)$$

$$\psi_i \leftarrow (1 - \eta\bar{\lambda}_{\mathcal{I}})\psi_i - \eta(f_{ui} - y_{ui})(p\phi_u + (1-p)M_y\phi_u) \quad (35)$$

Here $\bar{\lambda}_{\mathcal{U}}$ and $\bar{\lambda}_{\mathcal{I}}$ are regularization constants suitably adjusted with respect to the prediction accuracy (they are obtained by normalizing Eq(32) in terms of size). The learning rate η is annealed with a discount factor after each iteration, as suggested by [40].

Algorithm 1 Stochastic gradient descent learning

initialize $M_y = M_s = I$, random ϕ_u and ψ_i
for $t = 1$ **to** T **do**
  Stochastic gradient descent w.r.t. ϕ_u with M_s fixed.
  Update M_s
  Stochastic gradient descent w.r.t. ψ_i with M_y fixed.
  Update M_y
end for

The absolute-value link function $|\xi|$ is approximated by a smooth variant $(\alpha^{-1}\log 2\cosh\alpha\xi)$ with $\alpha = 100$. This approximation avoids non-differentiability in the objective function and aids convergence of the optimization.

6.3.3 Empirical Results for Unsupervised BRI

We conducted experiments on the Epinion data set. The social relationships in Epinion are dominated by positive ones (over 85% of the edges are positive). Following the settings of [46, 28], we derive a balanced dataset containing equal numbers of positive and negative relations by randomly down-sampling the positive edges. Moreover, because the behavioral data in this dataset is extremely skewed (the mean value of the ratings is around 4.6), we standardize all the ratings to the range $[-1,\ 1]$ by applying a transformation: $(y-3.0)^{+}-1.0$.

We evaluate the BRI model in terms of its unsupervised sign prediction accuracy. For this purpose, we hide the signs of all the relationships in the training set, i.e., all the edges are treated as "acquaintance" regardless whether they were originally trust or distrust. For comparison, we also evaluate the following three unsupervised methods:

- RANDOM. A random guessing predictor yields 50% accuracy because the dataset is balanced.
- CORRELATION. The 1-dimensional classifier:

  $$\hat{s}_{uv} = \text{sign}(\gamma_{uv} - \gamma_0)$$

  based on the behavioral correlations γ, where γ_0 is the optimal threshold found by line-search.
- FIP. The Friendship-Interest Propagation model [93] has been proved effective for link (tie-strength) prediction using both social and behavior data. FIP can be viewed as a special case of BRI with (1) both $p = 1$ and $q = 1$; and (2) no automatic sign determination.

Throughout this chapter, all the models are evaluated by using the 1-dimensional prediction scheme Eq(33). In particular, for the output score ρ_{uv} of a model, we predict the sign as

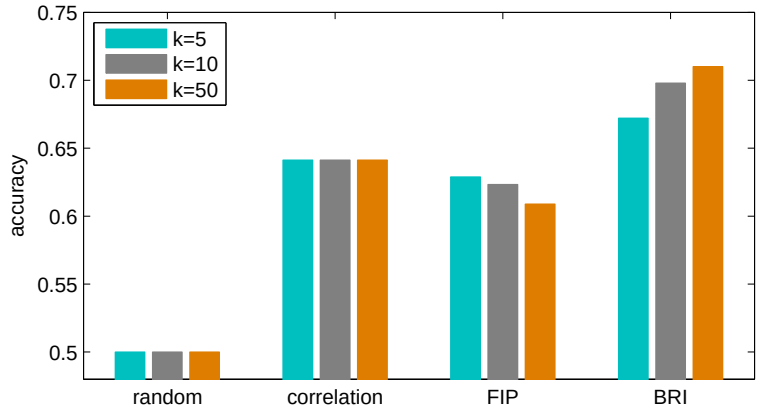


Figure 15: Unsupervised sign prediction performance on Epinion.

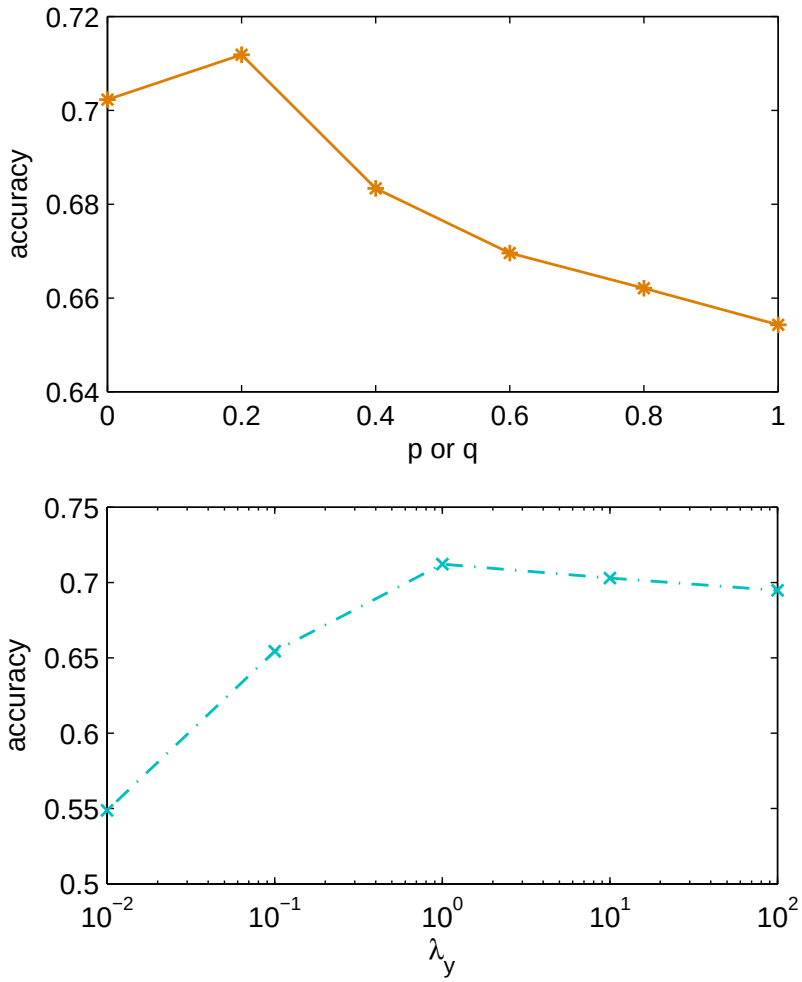


Figure 16: Unsupervised sign prediction accuracy as a function of the coloring proportion $p = q$ and the weight for behavioral data, λ_y .

Table 8: Semi-supervised sign prediction performance on Epinion data.

Model	RAND	FIP	CORR	LOO-LR	BRI						
Labels	0	0	0	All - 1	0	1%	5%	10%	20%	50%	90%
Accuracy	0.500	0.591	0.644	0.934	0.709	0.731	0.747	0.776	0.818	0.869	0.912

$\hat{s}_{uv} = \text{sign}(\rho_{uv} - \rho_0)$, where ρ_0 is an optimal threshold scalar found by line search on validation data.

The prediction accuracy is depicted in Figure 15, where the results of BRI and FIP are reported at latent dimensionality $k = 5$, 10 and 50 respectively. As can be seen from the figure, BRI achieves up to 71% accuracy in this unsupervised setting and it significantly outperform all the three baselines. In particular, BRI gains 42% improvement over RANDOM, 21% over FIP and 11% over CORRELATION. The result is quite promising especially considering that all the signs are hidden in training. It also proves the BRI model is potentially usable in practice for en-signing unsigned social networks, given that we can predict trust/distrust from acquaintance relationships with over 70% accuracy.

One key notion that motivates our work is that we assume the behavioral data should be useful for tie-sign prediction. As can be seen from Figure 15, using the behavioral data alone, the simple correlation-based predictor achieves the second best performance among all the four models. This observation validates our hypothesis and indicates that behavioral information is indeed highly predictive for signed social ties. In Figure 16 (b), we further report the accuracy of BRI as a function of the weight for behavioral data, i.e., λ_y in Eq(32). As we can see, while the best result is achieved at a moderate value of λ_y, interestingly, the performance does not degrade badly if we further increase λ_y. In contrast, if λ_y is decreasing toward 0, the performance degrades quickly towards that of random prediction — in an extreme, BRI trained on unsigned social data alone does not perform substantially better than RANDOM. These observations reveal that, in the unsupervised setting, the behavioral information is even more predictive than, and should be emphasized over, the topology of the acquaintance graph.

Note that although the FIP model also leverages the behavioral data, it is ineffective in sign prediction. FIP estimates the tie-strength between two users, i.e., it attempts to predict whether two users, u and v, *know each other*. Nonetheless, it is incapable to predict whether they *trust or distrust each other*, i.e. FIP cannot discern the sign of a connection. From another prespective, this comparison between FIP and BRI confirms that the proposed mechanism of *automatic sign determination* is effective for sign prediction.

The performance of the BRI model is affected by the parameters of mixture probabilities, i.e., p and q in Eq(29-31). In Figure 16, we illustrate how the prediction accuracy of BRI changes with these two parameters, where for simplicity p and q are set to the same values. As can be seen from the figure, the accuracy curve is typically in a inverted U-shape with the optimal performance achieved at around 0.2. This observation suggests that both the two groups of factors (i.e. intrinsic taste vs influence, prior impression vs behavior agreement) are important and should be taken into account simultaneously as what we do in the BRI model.

6.3.4 Semisupervised BRI

We now consider semi-supervised formulation, where part of the labels for social ties are observed — in practice, these labels could be obtained directly from users through an

online survey or simply by manual annotations. In particular, assume that in addition to the behavioral observations $\mathcal{O}_y$ and the acquaintance relations $\mathcal{O}_c$, we also have access to a small set of labeled relations (i.e., trust-distrust) on pairs $(u, v) \in \mathcal{O}_s$. In this case we add

$$\lambda_s \sum_{(u,v)\in\mathcal{O}_s} (s_{uv} - h_{uv})^2 \tag{36}$$

to the objective function of Eq(32). This formulation has a close connection with the semi-supervised classification model of [6]. It maximizes the fidelity to the *target output* (e.g. s_{uv}) of the labeled data, while approximating the target output for unlabeled data up to an arbitrary sign. A notable distinction is that we are learning latent factor models (e.g., the BRI model) rather than a global feature mapping (i.e., the classifier of the form $w^\top x + b$ in [6]).

The learning algorithm and prediction formula are similar to the unsupervised case and therefore are omitted.

6.3.5 Empirical Results for Semisupervised BRI

Table 8 gives the overall results of semi-supervised sign prediction on Epinion data, where we report the performance of the BRI model that are trained using progressive proportions (1% to 90%) of the labeled data. As a reference, we also report the results obtained by RANDOM guessing, CORRELATION-based 1-dimensional classifier, the Friendship-Interest Propagation (FIP) model of [93] and the leave-one-out logistic regression model (denoted LOO-LR) of [46].

Our first observation from the table is that the performance of the BRI model is approaching that of LOO-LR. As we increase the labeled data used for training, we see that the prediction accuracy of the BRI model is steadily improving, indicating that the labeled data can indeed be used to improve the prediction ability of BRI, somewhat as we expected. In particular, with 1% of labeled data, we gain over 3% improvement of accuracy; and if we use over 90% of the labeled data, BRI can achieve comparable performance as the leave-one-out prediction of [46] but using far less supervision information compared to the latter.

Interestingly, it seems that the labeled data are not always helpful for BRI. For example, the model trained using 100 (i.e. less than 0.05%) labeled edges does not perform significantly better (in some cases, even slightly worse) than the unsupervised model. This observation suggests that the amount of the labeled data should exceed a minimum threshold in order for it to take effect. This is quite different from the traditional semi-supervised learning in which labeled data, whenever available, seems to always help prediction. Nonetheless, it is reasonable in our context because we are learning a latent factor model over extremely large user/item spaces — the model is almost inevitably overparameterized and with a lot of local optima due to the nonconvexity; the labeled data cannot guarantee to drive the model toward a right direction if there is too little of it.

6.4 Encoding Social Psychology

Theories from social psychology [28] provide profound perspectives for understanding the formation of signed relations in social networks. According to [47, 46], satisfactory sign prediction accuracy could be obtained by applying social psychology principles. For example, [46] constructed local topology-based features according to status and balance theories

Table 9: Comparison of sign prediction performance on Epinion: models with vs without sociological principles.

%Labels	0%	1%	5%	20%	50%	90%
BAL	0.539	0.557	0.582	0.601	0.662	0.735
STA	0.617	0.719	0.757	0.799	0.824	0.843
BRI	0.709	0.731	0.747	0.818	0.869	0.912
BRI+BAL	0.711	0.734	0.751	0.821	0.870	0.912
BRI+STA	0.714	0.739	0.763	0.833	0.884	0.925

and employed a logistic regression classifier based on these features to predict the signed relationships. In a leave-one-out setting, their model achieves up to 93% sign prediction accuracy on Epinion data. In this section, we extend the basic BRI models to encode social psychological principles, especially the theories of status and structural balance.

6.4.1 Encoding Structural Balance Theory

The first theory we try to encode is that of *structural balance* [31, 47]. Roughly speaking it implies the intuition that "a friend's friend is a friend" and that "an enemy's enemy is a friend". Basically, it considers the balance of signs on a triad involving three users. It states that a triad is balanced if and only if it contains an odd number of positive edges.

BALANCE THEORY [31, 47]: *A triad (u,v,w) is balanced in either of the two cases: (1) if* $s_{uv} = 1$ *and* $s_{uw} = 1$*, then* $s_{vw} = 1$*; (2) if* $s_{uv} = -1$ *and* $s_{uw} = -1$*, then* $s_{vw} = 1$.

To encode the balance theory, we introduce an additional (1-dimensional) latent factor β_u for each user u. Unlike the latent profiles ϕ, β is only used to capture the structural balance of the network topology. In particular, we assume an additional component:

$$g_{uv} = \beta_u \beta_v, \tag{37}$$

The social tie is modeled using the following mixture-of-effect model:

$$h_{uv} = q_1 \phi_u^\top \phi_v + q_2 \phi_u^\top M_s \phi_v + q_3 \beta_u \beta_v, \tag{38}$$

where $q_1,\ q_2,\ q_3 \in [0,\ 1]$ and $q_1 + q_2 + q_3 = 1$.

The new mixture component encodes the status theory because it implies: if $g_{uv} > 0$ and $g_{uw} > 0$, then $g_{vw} > 0$ (i.e. a friend's friend is a friend); if $g_{uv} < 0$ and $g_{uw} < 0$, then $g_{vw} > 0$ (i.e. an enemy's enemy is a friend). This is a well known parametrization of structural balance. In fact, it matches the models found in collaborative filtering by matrix factorization, albeit with reduced dimensionality. Effectively we are setting aside a subspace of the latent factors exclusively for social interactions.

6.4.2 Encoding Status Theory

In the previous sections we implicitly viewed the social network as an *undirected* graph. Here, we extend the BRI model to encode status theory [28, 47], which was developed for directed networks. That is, from now on, the social ties are viewed as asymmetric relationships. Basically, status theory assumes that there exists a partial order over the user space such that positive edges only goes from low-status nodes to nodes with higher

status; therefore, the relationships are transitive, i.e., if $s_{uv} = +1$ and $s_{vw} = +1$ then $s_{uw} = +1$, and vice versa.

STATUS THEORY[28, 47]: *A positive directed link $s_{uv} = +1$ indicates that the head node u has a higher status than the tail node v; a negative link $s_{uv} = -1$ indicates that u has a lower status than v.*

To encode status theory we introduce a global parameter θ to capture the partial ordering of users. θ maps the latent user profile ϕ_u to a scalar quantity $\ell_u = \theta^\top \phi_u$, which reflects the corresponding user u's *social status*. According to status theory, we characterize social ties from u to v by modeling the relative status difference between user u and user v:

$$\ell_{uv} = \theta^\top(\phi_u - \phi_v) = \underbrace{\theta^\top \phi_u}_{=:\ell_u} - \underbrace{\theta^\top \phi_v}_{=:\ell_v}. \tag{39}$$

Note that status theory implies that the social tie is typically an antisymmetric relationship, i.e. if $s_{uv} = +1$, then $s_{vu} = -1$. This implication is, however, too strong. For example, in the Epinion case, only 2% of the bi-directional relationships are truly antisymmetric. To relax this assumption, we therefore use a mixture of effects model as in the previous section. This yields:

$$h_{uv} = q_1 \phi_u^\top \phi_v + q_2 \phi_u^\top M_s \phi_v + q_3(\ell_u - \ell_v). \tag{40}$$

Since ℓ_u, ℓ_v induce a total order on users it satisfies the requirements of social status theory: by transitivity $\ell_{uv} > 0$ and $\ell_{vw} > 0$ imply $\ell_{uw} > 0$.

6.4.3 Empirical Results

We evaluate the two extended BRI models in the semi-supervised setting and report the results in Tabel 9. As a reference, we also report the performance of the balance model and the status model on its own. As can be seen from the table, by encoding the status theory, we achieve results that are comparable with the leave-one-out prediction of [46], but we use far less supervision, i.e. observed tie labels. Also note that encoding the balance theory only leads to very marginal improvements. The latter is to be expected since adding a small subspace to an already existing multiplicative latent factor model will not change the expressive richness of the model significantly.

Among all the methods, the 1D balance model performs the worst, e.g., in the unsupervised case, it performs even much worse than FIP and the correlation-based approach. Somewhat surprisingly, the plain status model perform quite satisfactorily. In most cases, the accuracy scores achieved by the status model are very close to those obtained by the basic BRI model. This observation is consistent with the empirical insight reported in [47] that status theory turns to explain the signed relationship better. However, these results do not entirely invalidate the balance theory because we use a rank-1 model that could be too simplistic for any practical network topology.

It is worth noting that the status model provides a *status mapping* $\ell_u = \theta^\top \phi_u$ that embeds the users into a 1-dimensional space with the coordinates corresponding to users' *social status*. In Figure 17, we visualize the Epinion network by randomly sampling 0.1% edges as a bipartite graph based on this embedding. In particular, the upper and lower horizontal lines denote the embedding spaces for head nodes and tail nodes respectively; edges that match the status theory (i.e., $s_{uv} = +1$ while $\ell_u > \ell_v$ or $s_{uv} = -1$ while $\ell_u < \ell_v$)

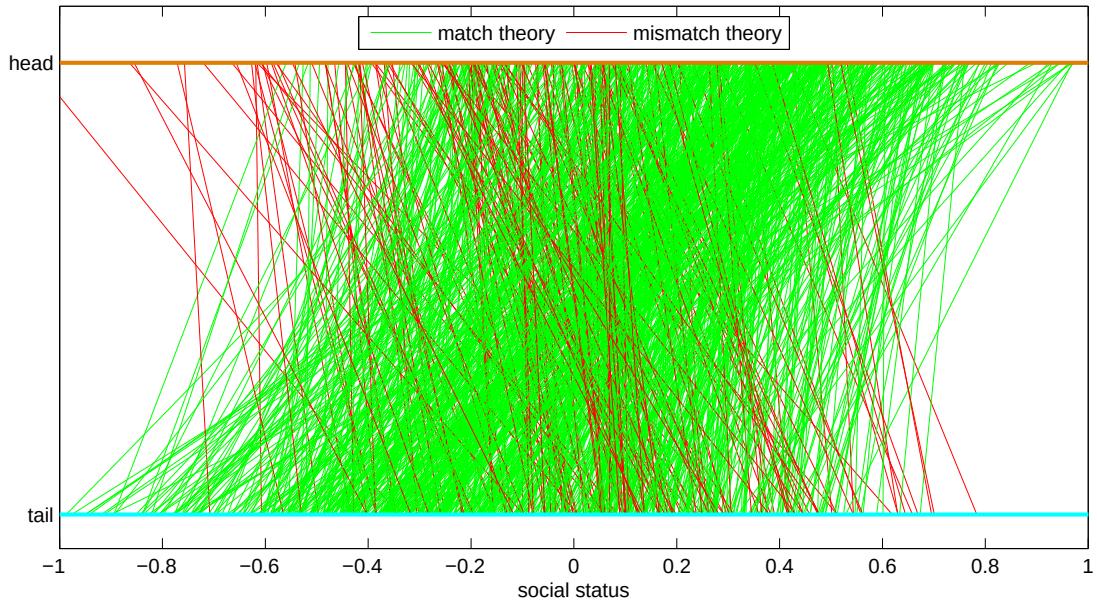


Figure 17: A visualization of social status and signed social ties for `epinion.com`.

Table 10: Behavior prediction performance on Epinion and Yahoo! Pulse.

	Epinion		Yahoo! Pulse		
Model	RMSE	MAE	AP@5	AR@5	nDCG@5
SVD	0.556	0.387	0.726	0.208	0.734
FIP	0.553	0.386	0.761	0.225	0.767
BRI	**0.524**	**0.350**	**0.778**	**0.232**	**0.785**

are plotted as green lines; in contrast, edges mismatch the theory are depicted as red lines; to give a clear illustration, negative edges are inverted, i.e., $s_{uv} = -1$ are replaced by $s_{vu} = +1$. This figure shows a clear picture about social status and signed relations. Over 84% of signed relations, i.e., trust/distrust, are indeed consistent with the theory, going from higher-status head nodes to lower-status tails. Noticeably, while there are confusions, they are mainly in the middle area between users with relatively small difference of social status. Also note that the confusions are becoming increasingly sparse for high-status head users, indicating that these users are more likely to be trusted by most other users, which is intuitively reasonable. An interesting investigation would be to examine how the social status discovered by this work relates to the well-known concept of *authority* [38] in social networks. We would like to leave such investigation for future research.

6.5 Behavior Prediction

A direct motivation of this work was to take advantage of signed relationships, when available, for behavior prediction purposes. The reasonable expectation in this context was that a stronger social signal would allow us to obtain better behavior prediction than what can be achieved via unipolar Homophily. In this section, we test whether, even in the absence of explicitly available signed social ties, we are able to improve behavior prediction estimates.

For this purpose, we compare three models: a plain collaborative filtering approach (SVD), the Homophily-based model (FIP) [93] and the proposed BRI model. We evaluate these models on both Epinion and Yahoo! Pulse data. For Epinion, we evaluate the rating prediction performance in terms of *root mean square error* (RMSE) and *mean absolute error* (MAE). For Pulse, because the behavior data is binary (either 1 or missing), we evaluate the top-k ranking performance, i.e. we compare the top suggestions of the model with the true actions taken by a user (i.e. whether he *liked* the item). We use the following three metrics commonly used in the IR community:

AP is the *average precision*. AP@k averages the precision of the top-k ranked list of each query (e.g. user).

AR or *average recall* of the top-k rank list of each query.

nDCG or *normalized Discounted Cumulative Gain* is the normalized position-discounted precision score.

In all three metrics we use $k = 5$ since most social networks use a similar number of items for service recommendation. For the Pulse data, we also implement the bias correction procedure suggested by [93].

The overall results are reported in Table 10. For both data sets we see that the BRI model consistently outperform the two baselines in all the measures. In particular, it improves the RMSE by 5.8% on Epinion, and the nDCG@5 score by 6.9% on Yahoo! Pulse. Note that BRI also significantly outperforms the FIP model. In terms of methodology, both FIP and BRI exploit social relations to improve behavior prediction; the difference, however, lies in how they leverage the social relations: FIP models the *unsigned tie-strength* between users which is then used to *reinforce* the decision making behavior of connected users; in contrast, BRI models the *signed ties*, distinguishing those whom we truly trust from those we do not. In this way, although behavior of two connected users is tied to each other, but not necessarily reinforcing. BRI therefore provides a mechanism that allows social users to follow the opinions from people they trust while at the same time fighting against the viewpoints of their foes. Our results suggest that signed social ties, such as trust-distrust relations, are a substantially more predictive signal in tying and regularizing social behavior than Homophily.

6.6 Summary

We examined the problem of predicting signed social ties, such as trust and distrust, based on the acquaintance relationships in social networks. This allows us to determine, with surprising accuracy, whether a link corresponds to a trustworthy friend or rather a frenemy. We present models that infer signed ties by capturing the interplay between social relations and users' behavior of decision making, and extend the models to encode general principles from social psychology. We investigate sign prediction in both unsupervised and semi-supervised settings, conduct experiments in a variety of perspectives and report promising results. We demonstrate that the predicted signed ties are much stronger signals for relating social behavior than traditional Homophily.

In the first two parts of the thesis, from Chapter 3 through Chapter 6, we have so far focused on examining behavior prediction and social contagion. Starting the next chapter, we move on to the third part of the thesis to examine user-generated contents.

CHAPTER VII

MINING USER COGNITIVE ASPECTS

Starting this Chapter, we move on to the third part of the thesis to examine user-generated contents especially texts. Users actively generate texts in social networks, for example, by posting a status update, commenting to a friend's activity, writing a note / miniblog on recent news, contributing a review to a newly released movie, retweeting to a hot topic, or simply asking a question or submitting a query for search, etc. As users in a social system are extremely diverse in their background (e.g., age, nationality, race, income), one important research topic is how to discover knowledge about the users from the textual contents they have generated. Even more interestingly, is it possible to discover some subtle and hidden properties of social users such as culture, knowledgability, personality, life-style, mood, etc. from social texts? Effective techniques developed for this purpose is not only useful for the industry to personalize social services, but also fundamentally important, for example, for privacy protection, cross-culture behavior comparison, etc. In this chapter, we demonstrate a model for identifying the technicality/knowledgeability of a user from the texts he has generated. One direct motivation of the work is to address the language gap between layman people and experts. In this chapter [91, 18], we seek to close the gap at the thematic level via *topic adaptation*, i.e., adjusting the topical structures for cross-domain documents according to a domain factor such as technicality. We present a probabilistic model for this purpose based on joint modeling of topic and technicality. The proposed τLDA model explicitly encodes the interplay between topic and technicality hierarchies, providing an effective topic-level bridge between lay and expert documents. We demonstrate the usefulness of τLDA with an application to consumer medical informatics.

7.1 Introduction

Although knowledge access is easier today than ever with the availability of numerous information sources on the Internet, transferring knowledge across domains remains a critical challenge. Particularly, transferring expert knowledge to lay users is hampered by the fundamental *language-gap* – lay users do not have enough literacy to understand expert jargons and terminologies; likewise, experts might be unfamiliar with the slang words to best popularize their expertise to, or precisely capture the inquiries of, a common audience.

Existing research [15, 100] attempts to close the gap at word-level by exploiting shallow word-correlations based on machine translation techniques, e.g., by augmenting or substituting the words in a lay document with a bundle of technical words that are statistically or semantically similar to the original text's content. These approaches are not entirely satisfactory because the translation is neither interpretable nor organized. They also turn to confuse different semantic themes as documents are assumed to be topically homogeneous throughout the corpus, which is, however, generally not true [10]. In this chapter, we attempt to close the gap at a deeper thematic level with a *topic bridge* that connects different domains semantically. We propose *topic adaptation*, a framework that adapts the underlying topical structures (rather than content words) according to a domain factor (e.g., time, sentiment, technicality) while the topics are discovered from cross-domain texts.

Topic adaptation naturally arises in cross-domain topic modeling. Probabilistic topic models [10, 26] interpret a document d as a mixture θ over a set of topical bases (multinomial distributions) β. A basic assumption in existing topic models is that all the documents within a corpus are drawn using the same *shared topical structures* (i.e., a single β). While this assumption works well for texts from a single domain, it is undesirable for texts from multiple domains where a significant language gap usually arises. For example, while lay texts in the topic "cancer" are dominated by common words like "cancer", "tumor" and "abnormal", an experts' knowledge base would favor technical words like "neoplasm", "carcinoma" and "metastasis". Naively learning the two domains with a single set of topical bases will inevitable lead to models with unacceptable fitting bias and in turn harm the quality of the extracted topics. Therefore, it is imperative to retain related but not identical topical bases β for each domain and capture the correlations among βs so that topics are both coherent within each domain and consistent across domains according to the changes of a domain factor (e.g., technicality). We refer to this problem as *topic adaptation* (TA).

The learning task involved in TA is challenging. On the one hand, separately learning topical bases β locally from each domain corpus will lose the topical correspondence among domains — there is no guarantee that the k-th topic learned from lay texts is thematically relevant to the k-th topic of expert documents; On the other hand, simultaneously learning multiple βs from the joint corpus requires decomposing text contents into multiple sets of word occurrence patterns (βs), which is intractable due to the lack of appropriate supervision – we only observe the words in each text but not their technicality stamps (i.e., the degree of a word being technical). Of course, the task would be greatly eased if the quantity of technicality, ideally for each word in the vocabulary, could be assigned a priori. However, manual annotation is often too expensive (e.g., vocabulary is huge, and dominated power-law by rare words that could easily exceed any individual's scope of literacy) and unreliable (e.g., any annotator could easily bias toward her own interest areas) to be practical.

In this chapter, we present the *topic-adapted latent Dirichlet allocation* (τLDA) for topic adaptation from cross-domain documents. The τLDA model devises a technicality-hierarchy in parallel to the topic-hierarchy of LDA, and encodes the interplay between the two hierarchies in the generative process. It leverages a mild supervision (the per-domain technicality stamps) to guide *cross-domain consistency*, making sure topics be adaptive to technicality changes. Moreover, it retains domain-specific topic bases βs to ensure *in-domain coherence*, which is efficiently parameterized via a two-mode mixture. We derive efficient inference and learning algorithm for τLDA based on variational Bayesian methods and evaluate it with an application to consumer medical informatics.

7.2 Related work

The language discrepancy between low-literacy laypeople and expert-produced documents has been widely recognized as a fundamental barrier for cross-domain knowledge transfer. [99, 73] observed that the language gap substantially degrades the performance of medical information services. [80] reported a similar challenge in legal informatics. Conventional efforts [42, 7] take a very manual approach, e.g., by educating clinicians and manually constructing communication scripts tailored for patients. Recently, several researches attempted to close this gap via word-level machine translation [15, 100]. In contrast, we attempt to bridge domains at the topic level to capture the deeper thematic correlation among domains, with add-on benefits such as readily interpretable results (e.g. the topic and technicality structures offer a comprehensible organization of texts for browsing or

summarization).

Topic models have been established for cross-domain texts, for example, the *cross-language topic models* [101, 63]. These works, however, are fundamentally different from ours: in their setting, topics are multinomials over different vocabularies; whereas in ours, topics are different multinomials over the same vocabulary. In essence, we are addressing the subtle variations within a language, which are arguably (more) challenging. These models are also limited in applications as they require a corpus containing approximately parallel documents. Perhaps most relevant to our work is the *dynamic topical model* [82], which learns drifted topics from time-evolving domains. Although such *topic drifting* is a special type of *topic adaptation*, the assumptions for *time factor* (for example, causality and Markov assumptions) are less suitable for other domain factors such as technicality.

Technicality is an important factor of natural language text, yet (surprisingly) rarely explored in topic modeling. A noticeable exception is the recently proposed *hierarchical topic model* or hLDA [8], which extracts a tree structure for learned topics. Although the depth of each topic in the hLDA topic tree roughly reflects the degree of its specificity, the only guidance for learning the tree is a nonparametric prior (i.e., the nested Chinese restaurant process), which admits a plausible *monotonic constraint* for topics: high-specificity topics are always contained by lower ones. Because the regulation imposed by the nCRP prior is rather weak and diminishing quickly as observations increase, and the monotone assumption could be inaccurate, the technicality quantified by hLDA is usually unsatisfactory. Furthermore, hLDA is not applicable to topic adaptation for cross-domain documents as it models a single topic structure β.

This chapter explores how the topic structure β evolves according to a continuous factor, e.g., technicality $\tau \in [0, 1]$. We do so by extracting a family of *aligned topic structures* $\{\beta(\tau)|\tau \in [0, 1]\}$, where for any topic ID k the multinomials $\beta_k(\tau_1)$ and $\beta_k(\tau_2)$ are semantically about the same theme (with different technicalities). Such aligned topic structures serve as a *topic-level bridge* which allow us to safely assess the similarity (e.g., match query with documents in IR) of two documents, d_1 and d_1, solely in terms of the corresponding topic memberships θ_1 and θ_2 without worrying how different their technicalities are or how they differ in BOW representations. The model is also practically appealing as it requires only mild corpus-level (rather than word-level) supervision.

7.3 Topic adaptation

Assume we have a family of domains $\{\mathcal{D}(\tau) : \tau \in [0, 1]\}$, distinguished from each other with distinct values of a domain factor[1] τ. Without loss of generality, we assume τ is a continuous variable in the range [0,1]. A domain is a collection of documents with the same τ, $\mathcal{D}(\tau) = \{(d, \tau_d) : \tau_d = \tau\}$, and a document is a finite sequence of words $d = w_1 w_2 \dots w_n$. Our goal in topic adaptation is to infer the topical structures and guarantee its in-domain coherence and cross-domain consistency for a given multi-domain corpus $\mathcal{D} = \bigcup_{\tau \in [0,1]} \mathcal{D}(\tau)$.

As in the *latent Dirichlet allocation* (LDA) model [10], we decompose the document-word co-occurrence matrix in terms of document-specific topic mixtures θ_d and a set of topical bases β (multinomial distributions). However, instead of a single common set of bases as in LDA, we retain domain-specific topic bases $\{\beta(\tau) : \tau \in [0, 1]\}$, which requires the nontrivial task of learning a functional family of multinomials. For simplicity, we adopt

[1]Hereafter, the domain factor τ always refers to *technicality*, although other factors such as *sentiment* and *time* might be equally applicable.

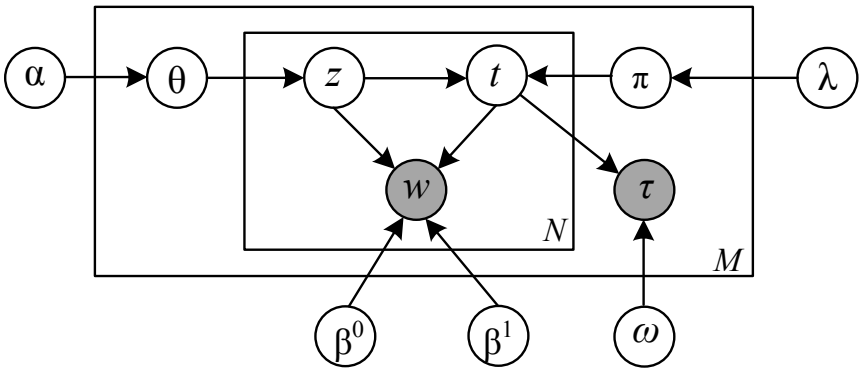


Figure 18: The topic-adapted latent Dirichlet allocation (τLDA) model.

a two-mode mixture to efficiently parameterize $\beta(\tau)$. Particularly, we assume any $\beta(\tau)$ is a mixture of two extremes $\beta^0 = \beta(0)$ and $\beta^1 = \beta(1)$, that is: $\beta(\tau) = (1-\tau)\beta^0 + \tau\beta^1$.

We establish a probabilistic generative model for topic adaptation. The key innovations are as follows: (1) we assume a hierarchy for technicality, in parallel to the LDA topic hierarchy; (2) we model the interplays between the topic and technicality hierarchies at the latent level; and (3) we let each word sampling be conditioned on both latent topic and latent technicality assignments. Specifically, the *topic-adapted latent Dirichlet allocation* (τLDA) model assume the following generation process for each document-technicality pair, (d, τ_d), in the joint corpus $\mathcal{D}$:

1. Draw topic mixture $\theta \sim$Dir(α)
2. For each topic, draw topic-level technicality $\pi_k \sim$Beta$(\lambda_{1k}, \lambda_{2k})$
3. For each word:
   a) Choose a topic assignment $z_n \sim$Mult(θ);
   b) Choose domain (i.e., technicality) assignment $t_n \sim$ Bernoulli(π_{z_n});
   c) Sample word $w_n \sim$ Mult$(\beta^{t_n}_{z_n})$;
4. Generate document technicality $\tau \sim p(\tau|t_{1:N}, \omega)$.

In the model, we assume the number of topics, K, is a priori specified and fixed (in practice, it could be determined via Bayesian model comparison [26]). As in the plain LDA model, the per-document topic mixture θ is drawn from a K-dimensional Dirichlet distribution Dir(α), and the per-word topic assignment z is from a discrete distribution conditioned on θ, i.e., Mult(θ). In parallel to this topic hierarchy, we also model a technicality hierarchy. The per-document technicality π is a K-vector, with each entry π_k specifying the degree of technicality for each topic; each π_k is drawn independently from a Beta distribution[2] Beta$(\lambda_{1k}, \lambda_{2k})$. The per-word technicality assignment t is a binary scalar, it is generated conditioned on both π and z from Bernoulli(π_z). Basically, when the i-th topic is sampled (i.e., $z_i = 1$) and its technicality π_i is given, t further specifies a domain – from which of the two extreme domains the topic is sampled (e.g., whether the "cancer" topic is talked about in layman or expert domain). Thereafter, a word is sampled conditioned on both topic and domain (technicality) assignment from a multinomial distribution Mult(β^t_z), where $\beta^t = (\beta^0)^{1-t}(\beta^1)^t$; both β^0 and β^1 are $K \times V$ matrices, where V is the size of the vocabulary. Finally, the technicality stamp associated with each document is modeled as

[2]In our implementation, we assume $\lambda_{1k} + \lambda_{2k} =$const for all k to further reduce parameters.

a response variable generated conditioned on all the technicality assignments: $p(\tau|\omega^\top \bar{y})$, where $\bar{y} = \frac{1}{N}\sum_{n=1}^{N} y_n$, $y_n = t_n z_n$ is a topic-aware code of t_n. For now, we use a cosine regression model that enjoys the best interpretability [88]; other models will be explored later. Particularly, we assume $p(\tau|\omega^\top \bar{y}) = \frac{1}{Z}\exp(\tau\omega^\top \bar{y})$, a degraded log-linear model with constant partition $Z = \int_0^1 \exp(\tau\omega^\top \bar{y})d\tau = \text{const}$. This model leads to regression by maximizing the Frobenius inner product between the model prediction and the ground-truth: $\omega = \arg\max\langle \tau_{1:M}, \omega^\top \bar{y}_{1:M}\rangle$.

The overall model is depicted with a graphical representation in Figure 18. For each (d, τ) pair, the joint distribution is given by:

$$\begin{aligned} P_d =& p(\theta, \pi, z_{1:N}, t_{1:N}, w_{1:N}, \tau|\alpha, \lambda, \beta^0, \beta^1, \omega) \\ =& p(\theta|\alpha)p(\tau|\omega^\top \bar{y}) \prod_{k=1}^{K} p(\pi_k|\lambda_{\cdot k}) \prod_{n=1}^{N} p(z_n|\theta)p(t_n|\pi_{z_n})p(w_n|\beta_{z_n}^{t_n}). \end{aligned} \tag{41}$$

7.4 Inference and learning

Both parameter estimation and inferential tasks in τLDA involve the intractable computation of marginalizing P_d over the latent variables. In this section, we derive approximate algorithms based on variational methods [34].

7.4.1 Variational approximation

We lower bound the log likelihood by applying the mean-field variational approximation:

$$\begin{aligned} \log p(d, \tau|\alpha, \lambda, \beta^0, \beta^1, \omega) &= \log \int_{\theta,\pi} \sum_{z,t} P_d \\ =& \mathcal{L}(\gamma, \Phi, \eta, \mu) + KL(q||p) \approx \max_{\gamma,\Phi,\eta,\mu} \mathcal{L}(\gamma, \Phi, \eta, \mu), \end{aligned}$$

where the posterior $p(\theta, \pi, \mathbf{z}, \mathbf{t}|d, \tau, \alpha, \lambda, \beta, \omega)$ is approximated by a variational distribution q. Here, we assume a fully-factorized distribution (per document) on the latent variables:

$$\begin{aligned} & q(\theta, \pi, z_{1:N}, t_{1:N}|\gamma, \phi, \eta, \mu) \\ =& \text{Dir}(\theta|\gamma) \prod_{k=1}^{K} \text{Beta}(\pi_k|\eta_{\cdot k}) \prod_{n=1}^{N} \text{Mult}(z_n|\phi_n)\text{Ber}(t_n|\mu_n) \end{aligned}$$

Denote $\mathcal{H}_q$ the entropy of q, ℓ the operator $\log p(\cdot)$, the variational lower bound (variational free energy) of the log likelihood, $\mathcal{L}$, is given by:

$$\begin{aligned} \mathcal{L} =& \mathbb{E}_q[\ell(\theta|\alpha)] + \sum_{n=1}^{N} \mathbb{E}_q[\ell(z_n|\theta)] + \sum_{k=1}^{K} \mathbb{E}_q[\ell(\pi_k|\lambda_{\cdot k})] + \mathcal{H}_q \\ &+ \sum_{n=1}^{N} \left(\mathbb{E}_q[\ell(t_n|\pi_{z_n})] + \mathbb{E}_q[\ell(w_n|\beta_{z_n}^{t_N})]\right) + \mathbb{E}_q[\ell(\tau|\omega^\top \bar{y})]. \end{aligned}$$

The terms in the first line are similar to those in LDA. The terms in the second line are given (in order) by:

$$\begin{aligned}
\mathbb{E}_q[\ell_{[t]}] &= \sum_{k=1}^{K} \phi_{nk} \left(\mu_n \Psi(\eta_{1k}) + (1-\mu_n)\Psi(\eta_{2k}) - \Psi(\eta_{1k}+\eta_{2k})\right) \\
\mathbb{E}_q[\ell_{[w]}] &= \sum_{k=1}^{K} \phi_{nk} \left(\mu_n \log \beta^1_{kv} + (1-\mu_n) \log \beta^0_{kv}\right) \\
\mathbb{E}_q[\ell_{[\tau]}] &= \frac{1}{N} \sum\nolimits_{n=1}^{N} \sum_{k=1}^{K} \tau \mu_n \omega_k \phi_{nk}
\end{aligned} \tag{42}$$

where v is the index of w_n in the vocabulary. By setting the derivatives of $\tilde{\mathcal{L}}$ (the Lagrangian relaxation of $\mathcal{L}$) w.r.t. the variational parameters to zero, we obtain the following coordinate ascent algorithm:

$$\gamma_k = \alpha_k + \sum\nolimits_{n=1}^{N} \phi_{nk} \tag{43}$$

$$\eta_{1k} = \lambda_{1k} + \sum\nolimits_{n=1}^{N} \phi_{nk}\mu_n \tag{44}$$

$$\eta_{2k} = \lambda_{2k} + \sum\nolimits_{n=1}^{N} \phi_{nk}(1-\mu_n) \tag{45}$$

$$\begin{aligned}
\phi_{nk} \propto\ & (\beta^0_{kv})^{1-\mu_n} (\beta^1_{kv})^{\mu_n} \exp\{\Psi(\gamma_k) - \Psi(\eta_{1k}+\eta_{2k}) \\
& + \mu_n \Psi(\eta_{1k}) + (1-\mu_n)\Psi(\eta_{2k}) + b_k \mu_n\}
\end{aligned} \tag{46}$$

$$\mu_n = \zeta\Big(\sum_{k=1}^{K} \phi_{nk}(\Psi(\eta_{1k}) - \Psi(\eta_{2k}) + \log \frac{\beta^1_{kv}}{\beta^0_{kv}} + b_k)\Big) \tag{47}$$

where $\Psi(\cdot)$ is the digamma function, the logistic mapping $\zeta(x) = \frac{1}{1+\exp(-x)}$, and b_k is a supervision bias due to the response model. For the cosine regression model, we have $b_k = \frac{1}{N}\tau\omega_k$.

These formulas are intuitively interpretable. We observe that the per-word topic distribution, ϕ, is learned as a result of negotiation between the two extreme domains. This can be seen by rewriting Eq(46) as $\phi_{nk} \propto (\phi^0_{nk})^{1-\mu_n}(\phi^1_{nk})^{\mu_n}$, where $\phi^i_{nk} = E_q[z_{nk}|t_n = i]$, $i = 0$ or 1. Particularly, each word occurrence is split according to its technicality into two parts, μ_n and $1-\mu_n$; then ϕ^0 and ϕ^1 are inferred individually in each domain conditioned on topic, domain, word as well as technicality samplings; and finally, the two domains negotiate with each other and output the combined results ϕ_{nk}. Another interesting observation is how the algorithm assigns technicality for each word. It uses a *logistic regression* model, where the per-topic regressors (consisting of three parts: the prior contrast $\Psi(\eta_{1k}) - \Psi(\eta_{2k})$, the domain contrast $\log \beta^1_{kv}/\beta^0_{kv}$, and the supervision bias b_k) are weighted by per-word topic distribution ϕ and then mapped by a logistic function.

We finally note that our variational inference algorithm for τLDA is efficient enough. From Eq(43-47), it requires $O(KN)$ operations per iteration, which is the same complexity as LDA.

7.4.2 Parameter estimation

The parameters of τLDA are learned by maximizing the evidence lower bound:

$$\max L = \sum_{m=1}^{M} \mathcal{L}_m(\alpha, \lambda, \beta, \omega; \gamma_m, \phi_m, \eta_m, \mu_m).$$

This two-layer optimization involves two groups of parameters, corresponding to τLDA and its variational model respectively. Optimizing alternatively between these two groups leads to a Variational Expectation Maximization (VEM) algorithm, where the E-step corresponds to applying variational approximation (i.e., Eq(43-47)) to each observation (d_m, τ_m) in the corpus and the M-step maximizes L with respect to the model parameters. Particularly, for the topic bases, we have:

$$\begin{aligned} \beta^0_{kv} &\propto \sum_{m,n,k} (1-\mu_{mn})\phi_{mnk} w^v_{mn}, \\ \beta^1_{kv} &\propto \sum_{m,n,k} \mu_{mn}\phi_{mnk} w^v_{mn}, \end{aligned} \tag{48}$$

where $w^v_{mn} = 1(w_{mn} = v)$, $1(\cdot)$ is the indicator function. From Eq(48), we see again that each word occurrence is split according to its technicality into two part, μ and $1-\mu$, which contribute to the two extreme topic bases β^1 and β^0 respectively.

Then, both the topic and technicality mixture priors, α and λ, are solved (independently) by Newton-Raphson procedures conditioned on values of γ and η respectively. And finally, the response parameter, ω is learned by maximizing the conditional likelihood:

$$\begin{aligned} &\max_{\omega} \mathbb{E}_q[\log p(\tau_{1:M}|\bar{y}_{1:M}, \omega)] = \langle \tau_{1:M}, \omega^\top \mathbb{E}_q[\bar{y}_{1:M}] \rangle, \\ &\text{s.t.} : ||\tau_{1:M}|| = ||\omega^\top \mathbb{E}_q[\bar{y}_{1:M}]||, \end{aligned} \tag{49}$$

where we pose a constraint to eliminate the scale freedom of ω. Based on the Karush-Kuhn-Tucker optimality of Eq(49), we derive a very simple close-form solution for ω:

$$\hat{\omega} = \bar{h}/||\bar{h}||_A,$$

where $\bar{h} = \frac{1}{M}\sum_m \tau_m \mathbb{E}_q[\bar{y}_m]$, $\mathbb{E}_q[\bar{y}] = \frac{1}{N}\sum_n \mu_n \phi_{nk}$, and $A = \mathbb{E}_q[\bar{y}_{1:M}]\mathbb{E}_q[\bar{y}_{1:M}]^\top / ||\tau_{1:M}||^2$, $||x||_A = \sqrt{x^\top A x}$ denotes the A-weighted l_2-norm.

7.4.3 Technicality analysis

Here, we derive empirical Bayesian methods to quantify technicality at different granularities. The first task is to predict the *document technicality*, which enables *domain identification* [89, 88]. For a given document d, we first run variational inference on d, then, we have: $\hat{\tau} = \omega^\top \bar{y}$. Note that the terms involving the supervision bias should be removed[3] in variational inference as τ is unobserved for incoming documents.

At a more compact level, *topic technicality* directly reflects the specificity of each topic, similar to the node-depth in the hLDA topic tree [8]. Here, we have:

$$\hat{\pi}_k = \mathbb{E}_d(\mathbb{E}_q[\pi_{mk}|d_m]) = \frac{1}{M}\sum_{m=1}^{M} \frac{\eta_{m,1k}}{\eta_{m,1k} + \eta_{m,2k}}. \tag{50}$$

Finally, *word technicality* analysis provides a function mapping for the vocabulary: $t(v) : \mathcal{V} \rightarrow [0, 1]$, which quantifies the relative specificity of a word w.r.t a targeted expertise-intensive domain and also the relative difficulty for a lay user to grasp. Again, we use empirical Bayesian:

$$\hat{t}_v = \mathbb{E}_d(\mathbb{E}_w[\mathbb{E}_q\{t_{mn}|w_{mn} = v\}|d_m]) = \sum_{m,n} w^v_{mn}\mu_{mn} / \sum_{mn} w^v_{mn}.$$

[3]For the cosine regression model, set τ_d=0.5 $\forall d$; for LR and LAD, set $b_k = 0$.

Table 11: The cross-domain corpus consisting of documents from five domains.

Domains	Yahoo!	PubMed	MeSH	CDC	WebMD
τ	0.0	1.0	1.0	0.7	0.3
#doc	74226	161637	25588	192258	275620

7.5 Experiments

In this section, we apply τLDA to medical documents. We wish to find how a same topic is expressed differently in lay and expert languages, and how topics are shifted according to domain technicality.

Data. As shown in Table 11, our corpus is a combination of documents collected from five different domains. The *Yahoo!* subset is a collection of user questions and corresponding answers from the health category of *Yahoo! QA* (`answer.yahoo.com`), representing lay domain labeled with lowest technicality (τ=0). The *PubMed* (medical journal articles from `www.pubmedcentral.nih.gov`) and *MeSH* (medical subject descriptors from `www.nlm.nih.gov/mesh`), in the other extreme, represent expert domain with highest technicality (τ=1). In between, *WebMD* (documents crawled from `www.webmd.com`) represents mildly non-technical domain (τ=0.3), and *CDC* (crawled from `www.cdc.gov`) mildly technical domain (τ=0.7)[4]. Note that these coarsely-assigned per-domain technicality labels, $\{\tau\}$, are the only pieces of supervision information we used for topic adaptation in τLDA.

Results The language gap leads to a substantial discrepancy of word usages between different domains, making it difficult to maintain a global vocabulary that is effectively balanced across domains (Otherwise, the vocabulary could be extremely skewed such that the majority of words come from lay domains). To this end, we first select terms (after stemming and stop-word removal) locally from each domain based on DF (document frequency) scores, and then interleave the sub-selection round-robin to form the global vocabulary (over 10K words).

An important issue for implementing τLDA is how to initialize β's. Although reasonable results are obtained by totally random initialization, we find that a simple pre-feeding initialization procedure leads to substantial performance improvement. Particularly, we first profile the technicality for each word by using the empirical average technicality of the training documents containing the word, i.e. $\hat{\mu}_v = \sum_{\{d: f_d^v > 0\}} f_d^v \tau_d$, where f_d^v denotes the term frequency of word v in document d; we then train a plain LDA model and compute the initial β's using Eq(48), where the pre-feeded word technicalities $\hat{\mu}$ are used in place of μ.

We train the τLDA model on a 60% subset of randomly sampled documents, and test on the rest. The results are averaged over 5 repeats. The variational algorithm is efficient: for each iteration, τLDA takes (on average) 7.6 more time than LDA (the LDA-C implementation) to converge.

Table 12 shows an intuitive view of six example topics found by τLDA. For each topic, we list the top-10 most probable words for lay (i.e., high values of β^0) and expert (i.e., high β^1) domains respectively. These results reveal a notable language gap between the

[4]WebMD is intended to be generally comprehensible, yet it contains substantially more technical words than Yahoo!. CDC is intended for both medical experts and public readers, more technical than average.

Table 12: Example topics found by τLDA: each topic is shown by the top-ten words in both layman domain (β^0) and expert domain (β^1); the top row indicates the technicality of each topic.

#1: $\pi = 0.06$		#2: $\pi = 0.15$		#3: $\pi = 0.18$		#4: $\pi = 0.19$		#5: $\pi = 0.26$		#6: $\pi = 0.54$	
β^0	β^1	β^0	β^1	β^0	β^1	β^0	β^1	β^0	β^1	β^0	β^1
who	protein	problem	activ	better	nucleic	how	relat	treatment	gene	recommend	data
ask	associ	risk	chemic	below	structur	think	program	weight	analysi	medicin	method
much	immunolog	you	process	she	same	you	report	your	determin	not	import
bodi	psycholog	your	therapi	children	inhibitor	someth	previous	food	blood	tell	deriv
you	purif	fill	substanc	farther	genom	femal	web	profession	model	littl	depart
not	virolog	thought	poison	abl	possibl	googl	various	fda	amino	past	diagnost
eat	enzymolog	skin	conserv	print	pcr	mmwr	file	health	enzym	test	measur
period	induc	anyth	organ	transmiss	express	histori	databas	diet	biosynthes	quit	complet
sometim	parasitolog	face	virus	season	chromosom	partner	establish	fat	signal	social	design
agre	patholog	regular	cell	treatment	dna	websit	analys	dose	yeast	progress	generat

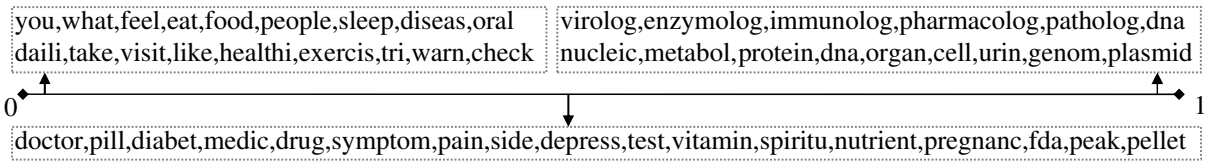


Figure 19: Example words with low-, medium- and high- technicalities.

two domains – almost all the representative words for lay domain are commonly-used or even slang words, in contrast, most words on the expert-domain side are highly-technical medical terminology (for example, words suffixed by "-ology"). The language gap is even evident when the same topic is concerned, indicating that laypeople and experts interpret differently even the same ideas.

As a reference, the technicality of each topic is also shown in the top row of Table 12. A very interesting observation is that there is no highly-technical topic — the maximum technicality for topics are around 0.5. In essence, this indicates that the language gap is asymmetric: experts can occasionally talk about topics of lay interests (but in a language mixing common words and their jargon), but laypeople are unlikely to be interested in expert's highly-technical topics. The absence of highly-technical topics in lay domain makes the corresponding word-occurrence pattern too submissive (infrequent) in the overall corpus to be captured by the model. To validate this hypothesis, we plot the topic distance between domains $D(\beta_k^0||\beta_k^1)$ (the Jensen-Shannon divergence) as a function of topic technicality π_k in Figure 20(a). As expected, we see an evident negative correlation between D and π, suggesting that the closer π_k is to the middle, the more β_k^0 and β_k^1 are overlapped. Also note that the technicality of a topic k does not depend on the word-distribution of β_k^0 or β_k^1, but rather on how much probability (i.e., relative frequency, see also Eq(50)) a topic is present in technical than lay domain. Therefore, although the first topic in Table 12 seemingly covers the most technical words in its β^1, it has the lowest technicality – it clusters very common word in β^0 and very-technical word in β^1, but the former appears far more frequent than the latter.

β^0 and β^1 are two different distributions over the same vocabulary. Because different

words have different intrinsic frequencies (e.g. technical words are less frequent), a better way to understand the learned topics might be to label each topic with most representative terms based on foreground-background contrast (e.g. by selecting words with highest ratio scores β^τ/β, where β is a background multinomial regardless of domains, $\tau \in \{0, 1\}$). According to this analysis, the two sets of topic bases are nicely aligned. Here, for ease of comparison with results of existing topic models, we comply with conventional topic labeling standard and demonstrate (in Table 12) each topic with most frequent words (i.e. solely according to foreground multinomial). Even from this somewhat naive analysis, we can still see that, except the first one, the two topical structures are approximately aligned. For example, topic #2 is about beauty and health, #3 birth and heredity, #4 medical records, #5 diet and weight control, #6 diagnosis and laboratory, etc. This observation indicates that, for a given k, β^0_k and β^1_k are roughly talking about the same topic. Such aligned topical bases are the key to cross-domain knowledge transfer. They fundamentally provide a topical bridge between lay domain and expert domain such that (1) documents from different domains can be mapped to the same simplex space $\mathcal{S} = \{\theta : ||\theta||_1 = 1, \theta_k \geqslant 0\}$, and (2) the distances between θ's precisely captures the semantic similarity between documents, no matter they are from same or different domains.

To quantitatively evaluate the quality of topic alignment, we perform an information retrieval task based on the topic mixture θ learned by τLDA. Our evaluation is confined by the availability of labeled data. As a preliminary test, we use a small number of lay documents as queries to retrieve technical documents. The results are manually graded on a 4-point scale ranging from 0 (irrelevant) to 4 (relevant). Based on this small data set of 25 queries with 100 documents per query [5], we report the performance in terms of the *normalized Discounted Cumulative Gain* on the top-five results (*nDCG@5*). The *nDCG@5* for τLDA is as high as 0.51 – a huge improvement over 0.38 of LDA based retrieval model [83].

τLDA also provide a simple mechanism to quantify technicality for words, which was previous achieved only by sophisticated models (e.g., hLDA). A rough view of word technicality learned by τLDA is given in Figure 19. We see that most results reasonably coincide with human intuition.

It would be interesting to examine the relationship between the topic bases learned by LDA (i.e., β) and those by τLDA (β^0 and β^1). For this purpose, we use an element-wise interpolation: $\beta_{kv} = x_{kv}\beta^1_{kv} + (1 - x_{kv})\beta^0_{kv}$ or $\log \beta_{kv} = x_{kv} \log \beta^1_{kv} + (1 - x_{kv}) \log \beta^0_{kv}$, and examine the distribution of the interpolation coefficient x. We find that the interpolations are distributed quite diversely: (1) a majority of β entries (about 68.3%, see also Figure 20(d)) are within the convex span of β^0 and β^1 (i.e., $x \in [0, 1]$), the rest 31.7% are not; (2) while the distribution peaks around $x = 0$ and $x = 1$, there is no singe dominant x that could fit all the entries well; (3) the correlation between $[x_{kv}]_{1:V}$ and $[t_v]_{1:V}$ is very low (in the range $[-0.05, 0.1]$), hence using a global per-word technicality function t_v to assist plain LDA (as in the initialization procedure) could not work either. These observations indicate that the interaction with technicality has fundamentally changed the topic structure so that from β to (β^0, β^1) is non-trivially a nonlinear decomposition. From another perspective, the observations also suggest that no *single* topic structure β is able to interpret a cross-domain corpus adequately well, hence, models with a unanimous β such as LDA will inevitably lead to substantial learning bias if brutally applied to this scenario.

[5] Labeled data for this task is extremely expensive as it requires annotators with moderate medical knowledge. We are working on collecting more labeled data from paid annotators and extending this experiment.

We also examined the TF difference (i.e., $\beta^1_{kv} - \beta^0_{kv}$ or $\log \beta^1_{kv} - \log \beta^0_{kv}$) between domains as a function of word-technicality t_v. Figure 20(b&c) shows the relationship for an example topic k, where the differences are normalized to [-1,1]. We see that the topic structures are nicely consistent with technicality: technical words are more frequent in technical domain than in lay domain, and vice versa.

We finally report the prediction performance of τLDA. Our first evaluation is based on the test-set log likelihood [81], a commonly used measure for topic models. We compare[6] τLDA with LDA and its supervised version (sLDA, [9]). The results are shown in Figure 20(e). We see that τLDA significantly outperforms both LDA and sLDA. This observation suggests that, by retaining topic bases for each domain, τLDA is more suitable for cross-domain topic learning than the other two competitors, which learn a single structure β for all the domains. We then apply τLDA to domain identification, i.e., to predict technicality τ_d for an unseen document d. Considering that our labeling of τ is very coarse (piecewise constant) and that precisely quantifying the degree of technicality for each domain is usually impractical in practice, this task requires a model capable of handling noisy data. The cosine regression model is too sensitive to noise to fulfill this purpose. Here, we consider two other response models. The first one is linear regression (LR):

$$p(\tau|\omega^\top \bar{y}) = \mathcal{N}(\tau|\omega^\top \bar{y}, \sigma^2);$$

The other is least absolute deviation (LAD):

$$p(\tau|\omega^\top \bar{y}) = \mathfrak{L}(\tau|\omega^\top \bar{y}, \delta),$$

where $\mathfrak{L}$ denotes the Laplacian distribution. For these two models, the inference algorithms are almost the same as that of cosine regression except that the supervision bias b_k is different. In particular, for LR:

$$b_k = \frac{1}{N\sigma^2}\tau\omega_k - \frac{1}{2N^2\sigma^2}[\sum_{i\neq n}\sum_j \omega_j\omega_k\phi_{ij}\mu_m + \omega_k^2].$$

For LAD, we have:

$$b_k = \text{sign}(\tau - \mathbb{E}_q[\bar{y}])\frac{\omega_k}{N\delta}, \text{ where } \mathbb{E}_q[\bar{y}] = \frac{1}{N}\sum_{nk}\omega_k\mu_n\phi_{nk}.$$

Similarly, the learning procedure is different only in estimating ω. Specifically, for the LR model:

$$\hat{\omega} = (Y^\top Y)^{-1}YT \text{ where } Y = \bar{y}_{1:M}\,,\, T = \tau_{1:M}.$$

The LAD regression leads to an *iterative reweighted least square* algorithm, which iterative updates:

$$\Lambda^{\text{new}} = \text{diag}(\hat{\omega}^{\text{old}\,\top} Y),$$
$$\hat{\omega}^{\text{new}} = (Y^\top \Lambda^{\text{new}} Y)^{-1}Y^\top \Lambda^{\text{new}} T.$$

The results are reported in Figure 20(f) with the *root mean squared error* (RMSE) as evaluation metric. The performance of τLDA with cosine regression is much worse than the others (RMSE>0.3) and is therefore omitted. We can see that, although sLDA is worse

[6]We use fold-in evaluation, e.g. for τLDA: $p(w_n|\mathcal{D}) = \sum_k \hat{\phi}_k(\hat{\mu}_n\hat{\beta}^1_{kv} + (1-\hat{\mu}_n)\hat{\beta}^0_{kv})$, where v is the ID of w_n in vocabulary. This comparison is fair across different models.

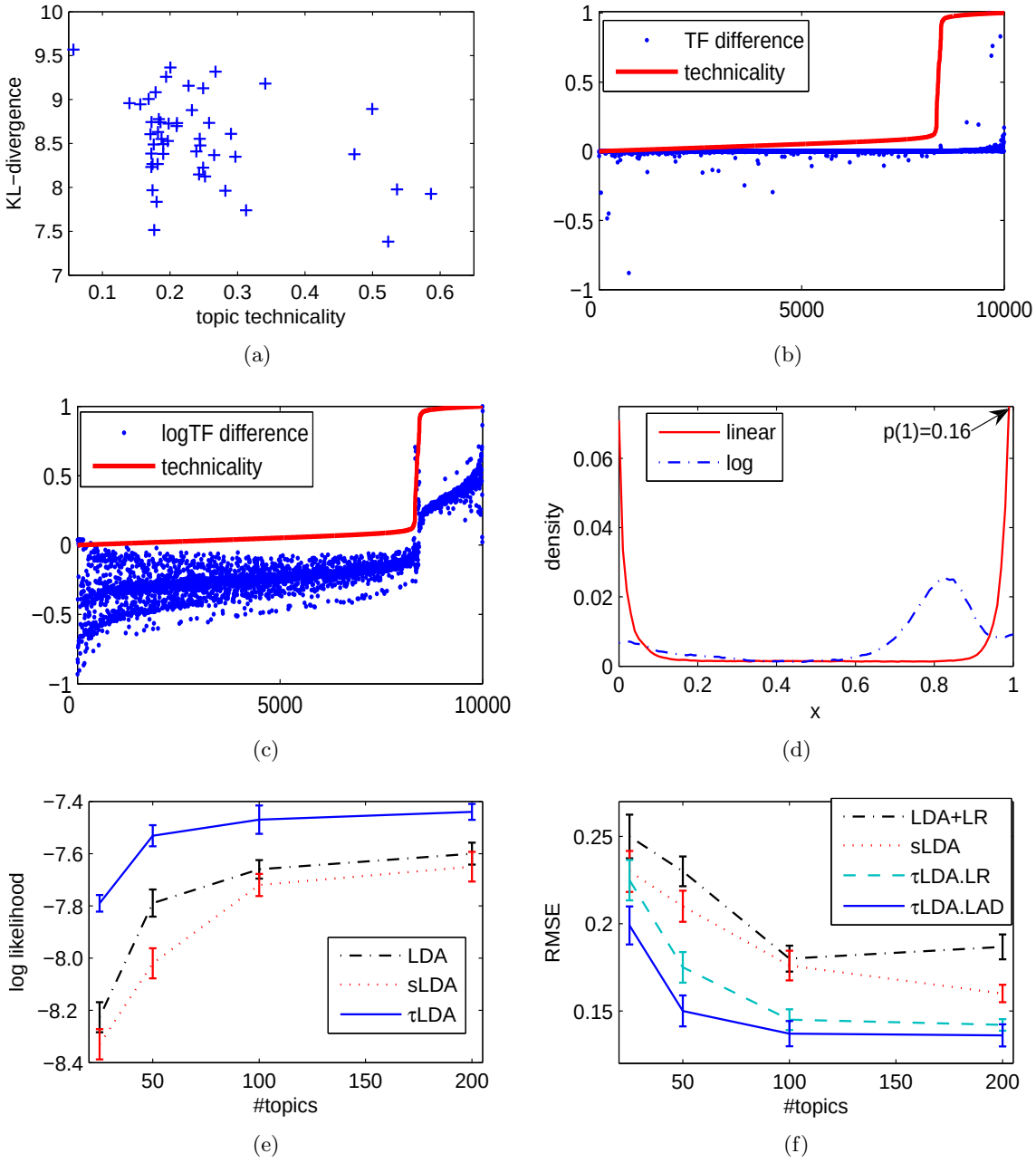


Figure 20: τLDA results on eHealth data: (a) topic variation vs. topic-technicality; (b-c) TF variation vs. word-technicality; (d) topic interpolation; (e) predictive likelihood; (f) domain identification accuracy.

than LDA in terms of predictive log-likelihood, it obtains better technicality prediction than LDA; yet, the two τLDA variants consistently outperform both LDA and sLDA (over 20% improvements). Also, less surprisingly, the LAD version of τLDA obtains significantly better performance than the LR variant as the former is more robust to noise.

7.6 Summary

We presented a generative model to learn related topic structures for documents from multiple domains. The τLDA model encodes both topic and domain factor (e.g., technicality) hierarchies as well as the interactions between them, providing an effective way to discover topic structures that are coherent within each domain and consistent among domains. The model offers a topic-level bridge for cross-domain knowledge transfer as demonstrated in eHealth tasks.

Today's personalized information services (e.g. Web 2.0) call for machine learning algorithms that are capable of capturing such subtle cognitive aspects of users (e.g. interests, capability, literacy, expertise, learning style) from their contextual texts and in turn adapting services accordingly. The τLDA offers a promising startpoint for learning user's literacy and expertise. It would be interesting to explore how other cognitive aspects of a user can be captured based on the texts she crafted/read.

CHAPTER VIII

CONCLUSION

The World Wide Web have brought to the public a new style of lives parallel to our day-to-day offline activities. The availability and scale of human activities on the Web raises tremendous opportunities and challenges for analytic study of human behavior. Effectively modeling of online human activities is not only of tremendous practical value to online industry, but also hold the key promise of deepening our understanding of human behavior and the society. Yet to date progress has been limited as the existing technologies still face severe challenges when applied to practice.

This dissertation aims to develop statistical models and machine learning algorithms for explanatory and predictive analysis of online human activities. We are particularly interested in three types of activities: decision making behavior, social interactions and user-generated contents. Our study accordingly focuses on three topics: behavior prediction, social contagion and content mining.

The first part of the dissertation is on behavior prediction, i.e., to predict users' online decision making behavior and to enhance the design of recommender systems. Recommender systems are commonly designed by analyzing the dyadic *user-item interactions* as can be recorded by a matrix, for example, users assigning ratings to movies. Research has thus been focused exclusively on estimating preference or equivalently completing the matrix. Nonetheless, our research shows that there are plenty of reasons to look at *user-system interactions* instead and formulate recommendation from a game-theoretic framework. In Chapter 3, we proposed a framework for learning user preference by modeling user choice behavior in the user-system interaction process. Instead of modeling only the sparse binary events of user actions as in traditional collaborative filtering, the proposed *collaborative-competitive filtering* models take into account the contexts in which user decisions are made and therefore are able to capture more accurate information about users' behavior. In Chapter 4, we further revisit the CCF preference model and present a novel game-theoretic framework for recommendation by viewing the user-system interactions at recommender system as buyer-seller interactions in a monopoly economic market. We show that the decisions of the user and the buyer are interdependent. This new perspective motivates us to optimize the action strategy of the system by first predicting users' reaction and then adapting its action to maximize the expected payoff in respect of certain strategic goals such as click-through rate, sales revenue and consumption diversity.

We further examine human activities in systems with a social networking functionality, where users are interacting with one another and *social contagion* plays an significant role. A social network is not solely a social graph that connect users with their friends, but it is also an interest graph that connect users with the resources (e.g., game, ad, brand) they like. One problem of fundamental interest is: how to effectively propagates both friendship and interest through the network. In Chapter 5, we exploited the well-established social effect of *Homophily* and proposed a framework for establishing an integrated graph that links a user to interested resources and connects different users with common interests. The proposed models estimate the tie-strength between two social users and in turn effectively propagates the right interest to the right person through the network. In Chapter 6, we

further investigated the more complicated *signed contagion*, where behavior of socially connected users could be reinforcing or conflicting. We examined the problem of predicting signed social ties, such as trust and distrust, based on the acquaintance relationships in social networks. This allows us to determine whether a link corresponds to a trustworthy friend or rather a frenemy. We presented models that infer signed ties by capturing the interplay between social relations and users' behavior of decision making, and extended the models to encode general principles from social psychology. Our results illustrated that the predicted signed ties are much stronger signals for relating social behavior than traditional Homophily.

The last part of the dissertation is on content mining, i.e., discovering knowledge from user-generated contents in online systems. We are particularly interested in seeking answers to three questions, i.e.: what are the intentions behind the contents? what do the contents suggest about the corresponding user? and what can we do to improve user experiences? In Chapter 7, we presented probabilistic models to automatically identify hidden cognitive aspects (e.g., knowledgeability) of a user from the texts he created. We particularly examined the knowledgeability of a user and the technicality of documents. We show the proposed τLDA model can be applied satisfactorily to address the language gap between layman people and experts, e.g., in health informatics. We also developed powerful models for text representation by presenting the *language pyramid* model [96] and the scale-space theory for text [98], which are particularly effective for short texts such as user-generated texts in online systems. We show the scale-space model can be used to address a variety of text mining tasks in a scale-invariant fashion. Moreover, we developed probabilistic models for ambiguous data [89, 88, 86] and demonstrated the effectiveness of these models in a variety of content mining tasks such as text classification, entity extraction, query disambiguation and image annotation. Note that the last two works are not included in this thesis, interested readers please refer to the papers for more details.

The long-term goal of my research is to establish a systematic paradigm for understanding, modeling and predicting online human activities and ultimately to help design more intelligent online systems. My future research will continue this line of research. In particular, my research will focus on the following dimensions: (1) *social dynamics*: how social relations, the network structure, and user's behavior of decision making (at both microscopic and macroscopic level) evolve over time; (2) *user cognitive aspect mining*: how to discover the subtle and hidden cognitive aspects of a user (e.g., culture, value, mood, habit, personality, knowledgeability) from the data s/he generated and exploit the knowledge discovered to enhance user experiences; (3) *multi-scale social analytics*: how social and behavioral phenomena scales, e.g., from individuals to communities to societies, from items to brands to companies, from daily to monthly to yearly; (4) *theoretic treatments of models* that provides guarantees and/or guidelines for model designing; and (5) algorithms and tools that scale the methodology developed up to meet the *big data* challenge.

REFERENCES

[1] Agarwal, D. and Chen, B.-C., "Regression-based latent factor models," in *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 19–28, ACM, 2009.

[2] Airoldi, E., Blei, D. M., Fienberg, S. E., and Xing, E. P., "Mixed membership stochastic blockmodels," in *NIPS '08: Advances in Neural Information Processing Systems 20*, pp. 33–40, 2008.

[3] Awerbuch, B. and Kleinberg, R., "Competitive collaborative learning," *Journal of Computer and System Sciences*, vol. 74, no. 8, pp. 1271–1288, 2008.

[4] Bakos, Y. and Brynjolfsson, E., "Bundling and competition on the internet," *Marketing Science*, vol. 19, no. 1, pp. 63–82, 2000.

[5] Balabanović, M. and Shoham, Y., "Fab: content-based, collaborative recommendation," *Commun. ACM*, vol. 40, pp. 66–72, March 1997.

[6] Bennett, K. P. and Demiriz, A., "Semi-supervised support vector machines," in *Advances in Neural Information Processing Systems 11*, vol. 11, pp. 368–374, 1999.

[7] Bickmore, T. W., Pfeifer, L. M., and Jack, B. W., "Taking the time to care: empowering low health literacy hospital patients with virtual nurse agents," in *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pp. 1265–1274, ACM, 2009.

[8] Blei, D. M., Griffiths, T. L., and Jordan, M. I., "The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies," *Journal of the ACM*, vol. 57, no. 2, pp. 1–30, 2010.

[9] Blei, D. M. and Mcauliffe, J. D., "Supervised topic models," in *NIPS '07: Advances in Neural Information Processing Systems 21*, 2007.

[10] Blei, D. M., Ng, A. Y., and Jordan, M. I., "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[11] Brynjolfsson, E., Hu, Y. J., and Simester, D., "Goodbye pareto principle, hello long tail: the effect of search costs on the concentration of product sales.," *Management Science*, Jan 2011.

[12] Brynjolfsson, E., Hu, Y. J., and Smith, M. D., "Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers," *MANAGEMENT SCIENCE*, vol. 49, pp. 1580–1596, November 2003.

[13] Brzozowski, M. J., Hogg, T., and Szabo, G., "Friends and foes: ideological social networking," in *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, (New York, NY, USA), pp. 817–820, ACM, 2008.

[14] Byrne, D. E., "The attraction paradigm," 1971.

[15] Can, A. B. and Baykal, N., "Medicoport: A medical search engine for all," *Computer Methods and Programs in Biomedicine*, vol. 86, no. 1, pp. 73–86, 2007.

[16] Chen, Y., Pavlov, D., and Canny, J. F., "Large-scale behavioral targeting.," in *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 209–218, ACM, 2009.

[17] Chu, W. and Park, S.-T., "Personalized recommendation on dynamic content using predictive bilinear models," in *WWW '09: Proceedings of the 18th international conference on World wide web*, pp. 691–700, ACM, 2009.

[18] Crain, S. P., Yang, S.-H., Zha, H., and Jiao, Y., "Dialect topic modeling for improved consumer medical research," in *Proceedings of the AMIA 2010 Annual Symposium*, pp. 132–136, 2010.

[19] Crain, S. P., Yang, S.-H., and Zha, H., "Understanding group dynamics in health forums," in *ICWSM '2012*, under review, Jan. 2012.

[20] Dumais, S. T., "Latent semantic analysis". in *Annual Review of Information Science and Technology*, 38: 188, 2005.

[21] Faloutsos, M., Faloutsos, P., and Faloutsos, C., "On power-law relationships of the internet topology," in *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pp. 251–262, ACM, 1999.

[22] Fleder, D. M. and Hosanagar, K., "Recommender systems and their impact on sales diversity," in *EC '07: Proceedings of the 8th ACM conference on Electronic commerce*, pp. 192–199, ACM, 2007.

[23] Gauch, S., Speretta, M., Chandramouli, A., and Micarelli, A., "User profiles for personalized information access," in *The Adaptive Web*, vol. 4321 of *Lecture Notes in Computer Science*, ch. 2, pp. 54–89, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[24] Gensch, D. H. and Recker, W. W., "The multinomial, multiattribute logit choice model," *Journal of Marketing Research*, vol. 16, no. 1, pp. 124–1328, 1979.

[25] Gilbert, E. and Karahalios, K., "Predicting tie strength with social media," in *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, (New York, NY, USA), pp. 211–220, ACM, 2009.

[26] Griffiths, T. L. and Steyvers, M., "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, pp. 5228–5235, 2004.

[27] Guadagni, P. M. and Little, J. D., "A logit model of brand choice calibrated on scanner data," *Marketing Science*, vol. 27, no. 1, pp. 29–48, 2008.

[28] Guha, R., Kumar, R., Raghavan, P., and Tomkins, A., "Propagation of trust and distrust," in *WWW '04: Proceedings of the 13th international conference on World Wide Web*, (New York, NY, USA), pp. 403–412, ACM, 2004.

[29] Gumbel, E. J., "Statistical theory of extreme values and some practical applications," in *Applied Mathematics Series*, vol. 33, National Bureau of Standards, 1954.

[30] Hausman, D. M., "Philosophy of economics", In *The Stanford Encyclopedia of Philosophy*, Fall 2008.

[31] Heider, F., "Attitudes of cognitive organization," *Journal of Psychology*, vol. 21, pp. 107–112, 1946.

[32] Herbrich, R., Graepel, T., and Obermayer, K., "Support vector learning for ordinal regression," in *In International Conference on Artificial Neural Networks*, pp. 97–102, 1999.

[33] Hofmann, T., "Probabilistic latent semantic analysis", In *UAI*, page 21, 1999.

[34] Jaakkola, T. and Jordan, M., "Bayesian parameter estimation via variational methods," *Statistics and Computing*, vol. 10, pp. 25–37, January 2000.

[35] Joachims, T., "A support vector method for multivariate performance measures," in *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pp. 377–384, ACM, 2005.

[36] Kallenberg, O., "Probabilistic symmetries and invariance principles,"

[37] Kameda, T., Ohtsubo, Y., and Takezawa, M., "Centrality in sociocognitive networks and social influence: An illustration in a group decision-making context," *Journal of Personality and Social Psychology*, vol. 73, pp. 296–309, August 1997.

[38] Kleinberg, J. M., "Hubs, authorities, and communities," *ACM Computing Surveys*, vol. 31, December 1999.

[39] Kondor, R. I. and Lafferty, J. D., "Diffusion kernels on graphs and other discrete input spaces," in *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 315–322, Morgan Kaufmann Publishers Inc., 2002.

[40] Koren, Y., "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *KDD '08: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434, 2008.

[41] Koren, Y., Bell, R., and Volinsky, C., "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[42] Kripalani, S., Jacobson, K. L., Brown, S., Manning, K., Rask, K. J., and Jacobson, T. A., "Development and implementation of a health literacy training program for medical residents," *Medical Education Online*, vol. 11, 2006.

[43] Kunegis, J., Lommatzsch, A., and Bauckhage, C., "The slashdot zoo: mining a social network with negative edges," in *WWW '09: Proceedings of the 18th international conference on World wide web*, (New York, NY, USA), pp. 741–750, ACM, 2009.

[44] Lafferty, J. and Lebanon, G., "Diffusion kernels on statistical manifolds," *Journal of Machine Learning Researc*, vol. 6, pp. 129–163, 2005.

[45] Lee, H., Raina, R., Teichman, A., and Ng, A. Y., "Exponential family sparse coding with applications to self-taught learning," in *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pp. 1113–1119, 2009.

[46] Leskovec, J., Huttenlocher, D., and Kleinberg, J., "Predicting positive and negative links in online social networks," in *WWW '10: Proceedings of the 19th international conference on World wide web*, (New York, NY, USA), pp. 641–650, ACM, 2010.

[47] Leskovec, J., Huttenlocher, D., and Kleinberg, J., "Signed networks in social media," in *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, (New York, NY, USA), pp. 1361–1370, ACM, 2010.

[48] Leskovec, J., Lang, K. J., and Mahoney, M., "Empirical comparison of algorithms for network community detection," in *WWW '10: Proceedings of the 19th international conference on World wide web*, pp. 631–640, ACM, 2010.

[49] Leyton-Brown, K. and Shoham, Y., *Essentials of Game Theory: A Concise, Multidisciplinary Introduction (Synthesis Lectures on Artificial Intelligence and Machine Learning)*. Morgan and Claypool Publishers, 1 ed., June 2008.

[50] Liben-Nowell, D. and Kleinberg, J., "The link prediction problem for social networks," in *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pp. 556–559, ACM, 2003.

[51] Liu, N. N. and Yang, Q., "Eigenrank: a ranking-oriented approach to collaborative filtering," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pp. 83–90, ACM, 2008.

[52] Long, B., Zhang, Z. M., Wu, X., and Yu, P., "Spectral clustering for multi-type relational data," in *International Conference on Machine Learning ICML*, pp. 585–592, 2006.

[53] Luce, R. D., "Individual choice behavior," 1959.

[54] Ma, H., Lyu, M. R., and King, I., "Learning to recommend with trust and distrust relationships," in *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, (New York, NY, USA), pp. 189–196, ACM, 2009.

[55] Ma, H., Yang, H., Lyu, M. R., and King, I., "Sorec: social recommendation using probabilistic matrix factorization," in *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pp. 931–940, ACM, 2008.

[56] Manski, C. F., "Maximum score estimation of the stochastic utility model of choice," *Journal of Econometrics*, pp. 205–228, August 1975.

[57] Markowitz, H. M., *Portfolio Selection: Efficient Diversification of Investments*. Wiley, 2 ed., Sept. 1991.

[58] Massa, P. and Avesani, P., "Trust-aware recommender systems," in *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, (New York, NY, USA), pp. 17–24, ACM, 2007.

[59] McFadden, D., "Conditional logic analysis of qualitative choice behavior," 1973.

[60] McLaughlin, M. R. and Herlocker, J. L., "A collaborative filtering algorithm and evaluation metric that accurately model the user experience," in *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 329–336, ACM, 2004.

[61] McPherson, M., Lovin, L. S., and Cook, J. M., "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[62] Miller, K., Griffiths, T., and Jordan, M., "Nonparametric latent feature models for link prediction," in *NIPS '09: Advances in Neural Information Processing Systems 22* (Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A., eds.), pp. 1276–1284, 2009.

[63] Mimno, D., Wallach, H. M., Naradowsky, J., Smith, D. A., and McCallum, A., "Polylingual topic models," in *EMNLP' 09: The 2009 Conference on Empirical Methods on Natural Language Processing*, pp. 880–889, ACL, 2009.

[64] Neumann, J. V. and Morgenstern, O., *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[65] Newman, M. E. J., "Detecting community structure in networks," *The European Physical Journal B - Condensed Matter and Complex Systems*, vol. 38, pp. 321–330, March 2004.

[66] Newman, M. E. J., "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, pp. 8577–8582, June 2006.

[67] O'Donovan, J. and Smyth, B., "Trust in recommender systems," in *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, (New York, NY, USA), pp. 167–174, ACM, 2005.

[68] Onnela, J. P., Saramäki, J., Hyvönen, J., Szabó, G., Lazer, D., Kaski, K., Kertész, J., and Barabási, A. L., "Structure and tie strengths in mobile communication networks," *Proceedings of the National Academy of Sciences*, vol. 104, pp. 7332–7336, May 2007.

[69] Owyang, J., "The many challenges of social network sites," *Web strategist blog*, Feb. 11 2008.

[70] Rosvall, M. and Bergstrom, C. T., "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, pp. 1118–1123, January 2008.

[71] Salakhutdinov, R. and Mnih, A., "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *ICML '08: Proceedings of the 25th international conference on Machine learning*, pp. 880–887, ACM, 2008.

[72] Sarwar, B., Karypis, G., Konstan, J., and Reidl, J., "Item-based collaborative filtering recommendation algorithms," in *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, ACM, 2001.

[73] Schwartzberg, Joanne G. and Vangeest, Jonathan B. and Wang, Claire, *Understanding Health Literacy: Implications For Medicine And Public Health*. American Medical Association Press, Dec. 2004.

[74] Shen, X., Tseng, G. C., Zhang, X., and Wong, W. H., "On psi-learning," *Journal of the American Statistical Association*, vol. 98, pp. 724–734, January 2003.

[75] Shmueli-Scheuer, M., Roitman, H., Carmel, D., Mass, Y., and Konopnicki, D., "Extracting user profiles from large scale data," in *MDAC '10: Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud*, (New York, NY, USA), pp. 1–6, ACM, 2010.

[76] Singh, A. P. and Gordon, G. J., "Relational learning via collective matrix factorization," in *Knowledge Discovery and Data Mining KDD*, pp. 650–658, 2008.

[77] Singh, A. and Gordon, G., "A unified view of matrix factorization models," in *ECML-PKDD '08*, vol. 5212, pp. 358–373, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[78] Smola, A. J. and Kondor, R., "Kernels and regularization on graphs," in *Proceeding of Annual Conference on Computational Learning Theory*, pp. 144–158, 2003.

[79] Tan, T. F. and Netessine, S., "Is tom cruise threatened? using netflix prize data to examine the long tail of electronic commerce," *Working paper, Wharton school of U Penn*, 2010.

[80] Uijttenbroek, E. M., Lodder, A. R., Klein, M. C. A., Wildeboer, G. R., Steenbergen, W. V., Sie, R. L. L., Huygen, P. E. M., and Harmelen, F. V., "Retrieval of case law to provide layman with information about liability: Preliminary results of the best-project," in *Computable Models of the Law: Languages, Dialogues, Games, Ontologies*, pp. 291–311, Springer, 2008.

[81] Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D., "Evaluation methods for topic models," in *ICML' 09: The 26th International Conference on Machine Learning*, pp. 1105–1112, ACM, 2009.

[82] Wang, C., Blei, D., and Heckerman, D., "Continuous time dynamic topic models," in *UAI' 08: the 24th Conference on Uncertainty in Artificial Intelligence*, 2008.

[83] Wei, X. and Croft, W. B., "Lda-based document models for ad-hoc retrieval," in *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 178–185, ACM, 2006.

[84] Weimer, M., Karatzoglou, A., Le, Q., and Smola, A., "Cofi rank - maximum margin matrix factorization for collaborative ranking," in *NIPS '07: Advances in Neural Information Processing Systems 20*, pp. 1593–1600, MIT Press, 2007.

[85] Weinberger, K., Dasgupta, A., Langford, J., Smola, A., and Attenberg, J., "Feature hashing for large scale multitask learning," in *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1113–1120, ACM, 2009.

[86] Xu, G., Yang, S.-H., and Li, H., "Named entity mining from click-through data using weakly supervised latent dirichlet allocation," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pp. 1365–1374, ACM, 2009.

[87] Xu, L., Neufeld, J., Larson, B., and Schuurmans, D., "Maximum margin clustering," in *Advances in Neural Information Processing Systems 17* (Saul, L. K., Weiss, Y., and Bottou, L., eds.), pp. 1537–1544, Cambridge, MA: MIT Press, 2005.

[88] Yang, S. H., Bian, J., and Zha, H., "Hybrid generative / discriminative learning for automatic image annotation," in *UAI '10: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 2010.

[89] Yang, S. H., Zha, H., and Hu, B.-G., "Dirichlet-bernoulli alignment: A generative model for multi-class multi-label multi-instance corpora," in *NIPS '09: Advances in Neural Information Processing Systems 22*, pp. 2143–2150, MIT Press, 2009.

[90] Yang, S.-H., "Local optimality of user choices and collaborative competitive filtering," *TR: arXive-1010.0622*, Oct. 2010.

[91] Yang, S.-H., Crain, S. P., and Zha, H., "Bridging the language gap: topic adaptation for documents with different technicality," in *AIStats '2011, JMLR W&CP 15*, pp. 823–831, 2011.

[92] Yang, S.-H. and Hu, B.-G., "A stagewise least square loss function for classification," in *SDM '08: Proceedings of the SIAM International Conference on Data Mining*, pp. 120–131, 2008.

[93] Yang, S.-H., Long, B., Smola, A., Sadagopan, N., Zheng, Z., and Zha, H., "Like like alike – joint friendship and interest propagation in social networks," in *WWW '11: Proceedings of the 20th international conference on World wide web*, ACM, 2011.

[94] Yang, S.-H., Long, B., Smola, A. J., Zha, H., and Zheng, Z., "Collaborative competitive filtering: learning recommender using context of user choice," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, SIGIR '11, ACM, 2011.

[95] Yang, S.-H., Smola, A. J., Long, B., and Zha, H., "Friend or frenemy? predicting signed ties in social networks," in *SIGIR '2012*, under review, Feb. 2012.

[96] Yang, S.-H. and Zha, H., "Language pyramid and multi-scale text analysis," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pp. 639–648, ACM, 2010.

[97] Yang, S.-H. and Zha, H., "Collaborative competitive filtering ii: Optimizing recommendation via collaborative games," *Work in progress*, Feb. 2012.

[98] Yang, S.-H. and Zha, H., "A scale-space theory for text," in *EMNLP '2012*, under review, Jan. 2012.

[99] Zeng, Q., Kogan, S., Ash, N., Greenes, R. A., and Boxwala, A. A., "Characteristics of consumer terminology for health information retrieval," *Methods of Information in Medicine*, vol. 41, no. 4, pp. 289–298, 2002.

[100] Zeng, Q. T., Crowell, J., Plovnick, R. M., Kim, E., and nd Emily Dibble, L. N., "Assisting consumer health information retrieval with query recommendations.," *Journal of the American Medical Informatics Association*, vol. 13, no. 1, pp. 80–90, 2006.

[101] Zhao, B. and Xing, E. P., "Bitam: bilingual topic admixture models for word alignment," in *ACL' 06: The 44th Annual Meeting of the Association for Computational Linguistics*, pp. 969–976, 2006.

[102] Zheng, Z., Zha, H., Zhang, T., Chapelle, O., Chen, K., and Sun, G., "A general boosting method and its application to learning ranking functions for web search," in *NIPS '08: Advances in Neural Information Processing Systems 20*, MIT Press, 2007.

[103] Zhou, D., Zhu, S., Yu, K., Song, X., Tseng, B. L., Zha, H., and Giles, C. L., "Learning multiple graphs for document recommendations," in *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pp. 141–150, ACM, 2008.

[104] Zhou, K., Yang, S.-H., and Zha, H., "Functional matrix factorizations for cold-start recommendation," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, SIGIR '11, pp. 315–324, ACM, 2011.