Broad Learning in Multiple Heterogeneous Domains

by

Chun-Ta Lu
B.S., Computer Science and Information Engineering, National Taiwan University, 2008
M.S., Computer Science and Engineering, National Chiao Tung University, 2011

Thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2018

Chicago, Illinois

Defense Committee:
Philip S. Yu, Chair and Advisor
Bing Liu, Professor at Dept. of Computer Science
Brian Ziebart, Assistant Professor at Dept. of Computer Science
Yuheng Hu, Assistant Professor at Dept. of Information and Decision Sciences
Xiangnan Kong, Assistant Professor at Worcester Polytechnic Institute

This thesis is dedicated to my parents

for their love, endless support,

and encouragement.

ACKNOWLEDGMENTS

First and foremost, I want to express my deepest gratitude to my Ph.D. advisor, Prof. Philip S. Yu, for his mentoring and endless support since 2012. He continually encouraged and convincingly conveyed a spirit of adventure in regards to research and scholarship. Without his successful guidance and persistent help along the way, I would not have the chance to freely explore a broad spectrum of research topics, and this dissertation would not have been possible.

I would like to thank Prof. Bing Liu, Prof. Brian Ziebart, Prof. Yuheng Hu, and Prof. Xiangnan Kong for taking their valuable time to serve as my dissertation committee members. I appreciate their insightful suggestions and crucial remarks that shaped this dissertation.

Additionally, I want to thank all my colleagues and friends that I met in the University of Illinois at Chicago. I am grateful to my colleagues for their collaboration and contribution in various projects related to this dissertation. I genuinely enjoy working with these great people, as well as countless delightful moments.

Nobody has been more important to me in the pursuit of the Ph.D. than the members of my family. I would like to thank my parents and sisters, whose love and guidance are with me in whatever I pursue. Even though they live far away from me, they offered me unconditional support and love for all these years.

CL

CONTRIBUTION OF AUTHORS

Chapter 1 is an introduction that outlines my dissertation research. Chapter 2 presents a published manuscript (Lu et al., 2014a) for which I was the primary author. Dr. Hong-Han Shuai, and my advisor Professor Philip S. Yu contributed to discussions about the preliminary ideas and assisted in revising the manuscript.

Chapter 3 presents a published manuscript (Lu et al., 2016), for which I was the primary author. Dr. Sihong Xie, and Professor Philip S. Yu contributed to the preliminary idea of transfer learning, and shared their suggestions on the experiments. Dr. Weixiang Shao, Dr. Lifang He contributed to the discussions with respect to the work and revising the manuscript.

Chapter 4 presents a published manuscript (Lu et al., 2017) for which I was the primary author. Dr. Lifang He, Dr. Weixiang Shao, Bokai Cao and Professor Philip S. Yu contributed to discussions with respect to the work and revising the manuscript.

Chapter 5 presents a published manuscript (Lu et al., 2018) for which I was the primary author. Dr. Lifang He, Hao Ding, Bokai Cao and Professor Philip S. Yu contributed to discussions with respect to the work and revising the manuscript.

TABLE OF CONTENTS

CHAPTER PAGE

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Recent years have witnessed the flourishing of heterogeneous data from various types of domains. For example, online review sites (like Amazon and Yelp) have access to contextual information of shopping histories of users, the reviews written by the users, as well as the description of the items. Broad learning is introduced to fuse such rich and complex heterogeneous information to improve the performance of learning tasks at hands.

In this dissertation, I will introduce our latest research progress on broad learning in multiple heterogeneous domains. In the first part, I focus on heterogeneous network based approaches for connecting and transferring knowledge across domains. To analyze the hidden connections and correlations between different domains, I present a methodology for identifying the same users in multiple domains (Lu et al., 2014a). I further propose a personalized recommendation algorithm that utilizes complementary information from related domains to improve recommendation performance (Lu et al., 2016).

To model the complex multi-way relationships among multiple tasks, in the second part, I propose a tensor-based framework for learning the predictive multilinear structure to solve multiple tasks altogether (Lu et al., 2017). Moreover, I present a generic method for learning structural data from heterogeneous domains, which can efficiently explore the high order correlations underlying relational structures of multi-way interactions (Lu et al., 2018).

CHAPTER 1

INTRODUCTION

1.1 Dissertation Outline

Nowadays, it is becoming common to have heterogeneous data obtained from various types of domains. For example, the visual information, tags, surrounding text, and user-generated annotations are often observed in multimedia data; the context of messages, friendships, and locations can be retrieved in social media. How to effectively incorporate such heterogeneous information is critical to good prediction performance for various applications, such as image classification, document categorization, and content recommendation. In particular, for newly-emerged domains, traditional supervised learning methods usually suffer from data-sparsity, as collecting sufficient labeled data is time-consuming and expansive. It is desirable to provide an effective broad learning solution to leverage discriminative information shared by related domains to help the learning task in the target domain, as well as improve the performance of each task altogether relative to learning them separately.

This dissertation focuses on the study of broad learning in multiple heterogeneous domains. Specifically, it contains two major approaches in broad learning: (i) heterogeneous network based approaches for connecting and transferring knowledge across domains in system level; (ii) matrix/tensor factorization based approaches for learning from multi-way interactions. Four different research directions related to the above two approaches are covered in this dissertation:

- In order to extract common information shared in related networks (domains in system level), we judiciously fuse the context features and network-based features to connect multiple heterogeneous networks.

- We provide a personalized recommendation model that can transfer knowledge from related domains to fit the newly-emerged domain, by capturing the rich similarity semantics based on both linkage structures and relationship attributes.

- To model the complex multi-way relationships between multiple tasks in multiple heterogeneous domains, we propose a tensor-based framework for learning the predictive multilinear structure to solve multiple tasks in the same time.

- We introduce a generic method for learning structural data from heterogeneous domains, which can efficiently explore the high order correlations underlying relational structures of multi-way interactions.

1.2 Connecting Heterogeneous Domains

(Part of this chapter was previously published in (Lu et al., 2014a).)

Connecting heterogeneous domains via identifying the entities (e.g., users and items) in common is promising to deal with the data-sparsity problem. In particular, personal social networks are considered as one of the most influential sources in shaping a customer's attitudes and behaviors, while they are barely observable in most e-commerce sites. In Chapter 2, we study the problem of customer identification in social networks, i.e., connecting customer accounts at e-commerce sites to the corresponding user accounts in online social networks such as Twitter. A novel method is introduced for identifying customers in online social networks effectively by using the basic information of customers, such as username and purchase history. It consists of two key phases. The first phase constructs both the context features as well as network-based features that can be used to compare the similarity between pairs of accounts across networks with different schema (e.g. an e-commerce company and an online social network). The second phase identifies the top-K maximum similar and stable matched pairs of accounts across partially aligned networks.

1.3 Transfer Learning for New Domains

(Part of this chapter was previously published in (Lu et al., 2016))

Transfer learning from related domains through overlapping entities is promising while it may be biased to the overlapping portion. Recent works suggest that preserving intrinsic closeness between similar entities is helpful to make the transfer learning robust. Most of the current similarity measures are computed by either comparing the relationship attributes between entities or leveraging linkage structures between entities. However, the former measures are non-trivial to be derived with insufficient information, and the performance of latter measures is often degraded because relationship attributes are discarded. Taking both the linkage structures and the augmented relationship attributes into account, in Chapter 4, we introduce a novel similarity measure, *AmpSim* (Augmented Meta Path-based Similarity). By traversing between heterogeneous networks through overlapping entities, AmpSim can easily gather side information from other networks and capture the rich similarity semantics between entities. We further incorporate the similarity information captured by AmpSim as graph regularization in a collective matrix factorization model such that the transferred knowledge can be iteratively propagated across networks to fit the new domain.

1.4 Multi-Task Multi-View Learning

(Part of this chapter was previously published in (Lu et al., 2017).)

Many real-world problems, such as web image analysis, document categorization and product recommendation, often exhibit dual-heterogeneity: heterogeneous features obtained in multiple views, and multiple tasks might be related to each other through one or more shared views. To address this Multi-Task Multi-View (MTMV) problems, in Chapter 4, we propose a tensor-based framework for learning the predictive multilinear structure from the full-order feature interactions within the heterogeneous data. The usage of tensor structure is to strengthen and capture the complex relationships between multiple tasks with multiple views. We further develop efficient multilinear factorization machines (MFMs) that can learn the task-specific feature map and the task-view shared multilinear structures, without physically building the tensor. In the proposed method, a joint factorization is applied to the full-order interactions such that the consensus representation can be learned. In this manner, it can deal with the partially incomplete data without difficulty as the learning procedure does not simply rely on any particular view. Furthermore, the complexity of MFMs is linear in the number of parameters, which makes MFMs suitable to large-scale real-world problems.

1.5 Modeling Multi-View Relational Data

(Part of this chapter was previously published in (Lu et al., 2018))

Different views may exhibit pairwise relations (e.g., the friendships between users) or even higher-order relations (e.g., a customer write a review for a product) among entities (such as customers, products, and reviews), and can be represented in a multi-way data structure, i.e., tensor. In Chapter 5, we introduce a multi-tensor-based approach that can preserve the underlying structure of multi-view relational data in a generic predictive model. Specifically, we propose structural factorization machines (SFMs) that learn the common latent spaces shared by multi-view tensors and automatically adjust the importance of each view in the predictive model. Furthermore, we provide an efficient method to avoid redundant computing on repeating patterns stemming from the relational structure of the data, such that SFMs can make the same predictions but with largely speed up computation.

CHAPTER 2

CONNECTING HETEROGENEOUS DOMAINS

(This chapter was previously published as “Identifying Your Customers in Social Networks”, in CIKM ’14 (Lu et al., 2014a). DOI: `http://dx.doi.org/10.1145/2661829.2662057`.)

2.1 Introduction

Personal social networks affect the adoption of individual innovations and products (Guo et al., 2011). For example, customers usually gather information from friends, when they contemplate purchasing goods and services. Customers also share opinions within their social networks regarding to different products which they have recently purchased or they are familiar with. Such actions of acquiring and disseminating information are critical to understanding customer behaviors and analyzing the factors that affect a customer’s decisions (Jiang et al., 2012a). However, these actions are implicit in the social connections (e.g., the relationship of friends or colleagues) that are barely observable in most e-commerce sites.

Fortunately, the emergence of online social networks, such as Twitter and Facebook, presents a great opportunity to access publicly available information of social connections. It appears that considerable potential exists for novel applications via leveraging the rich information from online social networks. Examples of applications include prediction of product adoption (Bhatt et al., 2010), personalized product recommendation via exploiting social correlation (Bhatt et al., 2010; Chua et al., 2013) , and maximization of product adoption and profits over social networks (Bhagat et al., 2012). In addition, mining the integrated information from social networks and commercial companies leads to other promising applications, such as discovering community of customers and analyzing opinions (Hu and Liu, 2004; Tsytsarau et al., 2013) of target customer communities for designing a marketing strategy. One common and crucial assumption of these applications is the knowledge of social connections between customers. However, since an online social network is built for social communication, this knowledge has not been used for e-commerce.

To fulfill the gap between conventional companies and social networks, in this chapter, we tackle the problem of customer identification in social networks. The mapping of customers to their user accounts in social networks serves as a prerequisite for applying existing marketing techniques to a broader range of e-commerce. Moreover, astroturfing becomes a serious problem in e-commerce nowadays. In 2013, the survey conducted by Dimensional Research[1] shows that 90% of consumers are influenced by online reviews in their purchasing decisions. The false advertising not only influences a large amount of customers to make wrong purchasing decisions but also slanders good products/companies. However, it is challenging to verify whether the review is spam or not due to lack of user information. Therefore, identifying customers in online social networks also provides a promising way to facilitate fake review detection[2].

[1] http://www.zendesk.com/resources/customer-service-and-lifetime-customer-value

[2] The privacy issues are worth discussing. According to the Consumer Privacy Bill of Rights, e-commerce sites should provide the privacy settings that allow users to avoid being tracking and keep their feedbacks/reviews private. On the other hand, users are encouraged and have better to adjust the privacy settings to their comfort levels.

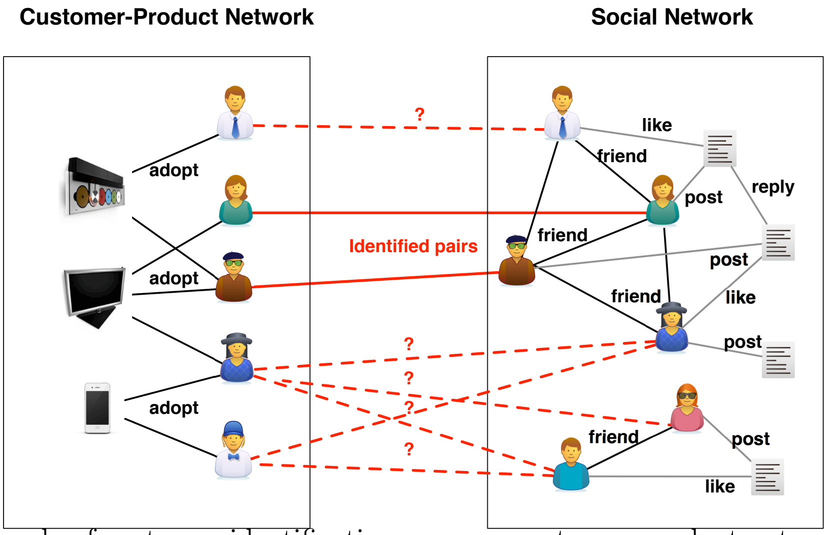


Figure 1. Example of customer identification across a customer-product network and a social network.

Generally, in an e-commerce system, customers interact with products (or services) only[1], while users in an online social network have connections with each other and interact with user-generated contents (e.g., tweets, pictures and videos posted by users). Therefore, the schema of these two systems are essentially different: the former is a bipartite customer-product network, but the latter is a general heterogeneous social network involving all kinds of connections among users and user-generated contents.

Figure 2.1 shows an example of a customer-product network and a social network. In the customer-product network, five customers adopt three products; meanwhile, six users discuss these products in the social network. Note that among the five customers, four of them also

[1] Although contents generated by customers are useful, they are rare in most commercial companies, and thus they are not included in this work.

TABLE I

Summary of related problems

Property	**Customer Identification**	**Anchor Link Prediction**	**Network Alignment**	**User Profile Matching**	**Relational Entity Resolution**	**Link Prediction**
reference		(Kong et al., 2013b)	(Bayati et al., 2013)	(Zafarani and Liu, 2013)	(Bhattacharya and Getoor, 2007)	(Getoor and Diehl, 2005)
target link relationship	one-to-one	one-to-one	one-to-one	one-to-one	clustering	many-to-many
target link type	inter-network	inter-network	inter-network	inter-network	intra-network	intra-network
#network	multiple	multiple	multiple	multiple	single	single/multiple
network schema	bipartite vs. heterogeneous[1]	heterogeneous	homogenous	heterogeneous	homogenous/ heterogeneous	homogenous/ heterogeneous
target network relationship	partially aligned	fully aligned	fully aligned	N/A	N/A	N/A

have user accounts in the social network but only two customers are identified (pairs of accounts marked in solid red lines). The task of customer identification is to discover which pair, as marked with question in Figure 2.1, belongs to the same individual.

Although users may create alias accounts on social networks, in most cases users will stick to a single account because of the difficulty of managing multiple accounts. Furthermore, only the primary account that reveals the major social activities is of interest to the investigation. Hence, we assume that each customer shall be identified as at most one (primary) user account in social network and vise versa.

Despite its value and significance, the customer identification task has not been addressed as it is very challenging due to the following two reasons:

1) *Difference in network schema.* Unlike most prior works on link prediction (Getoor and Diehl, 2005; Liben-Nowell and Kleinberg, 2003; Lichtenwalter et al., 2010; Kong et al., 2013b), customer identification requires to predict links across networks with completely different schema (i.e., bipartite network vs. general heterogeneous network). Most existing features

for link prediction, such as number of common neighbors and Jaccard's coefficient, are computed by enumerating the connections between nodes within a single network. However, due to the one-to-one constraint on the links across multiple networks, existing features will reduce to a constant value if we directly apply them to predict links across networks (Kong et al., 2013b). The situation is even more severe when one of the networks is a bipartite network, where no connections exist between customers. Although a bipartite network can be projected onto a unimodal network (Benchettara et al., 2010), such as a co-adoption network, many important features (e.g., interests of customers) will be lost during the transformation. Furthermore, customers barely have social interactions with neighbors in the unimodal network (Crandall et al., 2008).

2) *Partially aligned networks.* Another fundamental problem lies in the fact that most networks can only be partially aligned, w.r.t the one-to-one constraint. For example, in Figure 2.1, not all customers have accounts in the social network. Thus, anchor link prediction (Kong et al., 2013b) and conventional network alignment approaches (Bayati et al., 2013), which assume that two networks are fully aligned, cannot be directly used in the customer identification problem. A detailed comparison between customer identification problem and other related problems are reported in Table I.

To tackle the customer identification problem involving the above issues, we present the following contributions:

- We formulate the customer identification problem and present the problem analysis. To the best of our knowledge, our work is the first to focus on connecting users between e-commerce companies and online social networks (Section 2.2).
- Our approach, called CSI (Customer-Social Identification), can be applied to most e-commerce companies by using the basic information of customers, such as username and purchase history. To compare the similarity between users across networks, we transform existing social features for link prediction into heterogeneous features, e.g., common interests of users across networks (Section 2.3.1).
- We propose to formulate the multi-network partial alignment problem as a top-K maximum similarity and stable matching problem. Based on scores of similarity, CSI method can effectively identify customers in social network w.r.t one-to-one constraint (Section 2.3.2).
- Through extensive experiments on real-world datasets spanning 10 months, we demonstrate that CSI method consistently outperforms other commonly-used baselines – with up to 38% improvement on F1-score and 21% improvement on AUC (Section 2.4).

2.2 Problem Formulation

The customer identification problem we focus on, in this chapter, is to connect customer accounts at an e-commerce site (represented as a customer-product network) to the corresponding user accounts in an online social network. Though the proposed framework can easily be generalized to the setting with more than one pair of networks. In this section, we first define the concept of customer-product network and social network, and then present the formulation

TABLE II

Notation Summary

Symbol	Definition and Description
$\mathcal{G}^s$	network $\mathcal{G}^c = (\mathcal{U}^c, \mathcal{P}, \mathcal{E}^c)$, $\mathcal{G}^s = (\mathcal{V}^s, \mathcal{E}^s)$
$\mathcal{V}^s$	set of nodes in $\mathcal{G}^s$, $\mathcal{V}^s = \mathcal{U}^s \bigcup \mathcal{P}$
$\mathcal{U}^s$	set of users in $\mathcal{G}^s$
$\mathcal{P}$	set of products in both $\mathcal{G}^c$ and $\mathcal{G}^s$
$\mathcal{E}^s$	set of edges in $\mathcal{G}^s$
$\mathcal{A}$	set of the identified pairs across networks. $\mathcal{A}^*$ is the optimum set
$\Gamma(v_i^s)$	neighbors of the node v_i in $\mathcal{G}^s$
$\Gamma_u(u_i^s)$	friends of the user u_i in $\mathcal{G}^s$
$\Gamma_p(u_i^s)$	products that link to user u_i in $\mathcal{G}^s$
$\Gamma_u^s(p_x)$	users that link to the product p_x in $\mathcal{G}^s$
u_i^s	user in $\mathcal{G}^s$
p_x	product in $\mathcal{G}^c$ and $\mathcal{G}^s$
(u_i^c, u_j^s)	candidate pair across networks
$f(u_i^c, u_j^s)$	customer identification function
$score(u_i^c, u_j^s)$	similarity score of candidate pair (u_i^c, u_j^s)

of the customer identification problem. Table II lists the main notations we use throughout the chapter.

Customer-Product Network: Let $\mathcal{G}^c = (\mathcal{U}^c, \mathcal{P}, \mathcal{E}^c)$ denote a customer-product network, where $\mathcal{U}^c$ is the set of customers, $\mathcal{P}$ is the set of products, and $\mathcal{E}^c \subset \mathcal{U}^c \times \mathcal{P}$ is the set of *adoption links*. The type of adoption, depending on the genre of the e-commerce site, can be purchase of a product, subscription of a video or check-in on a hotel. To provide a general model for most e-commerce sites, we consider only the structure properties between customers and products and discard the semantic meaning of the adoption.

In online social networks, a large amount of contents is generated by users daily and most of them are irrelevant to the concerns. For the sake of efficiency, one may filter out redundant messages by setting predefined rules. For instance, an e-commence company can specify a list of terms related to the products of interest to the company and inquire for the relevant contexts from online social networks. Therefore, the user-generated contents in the social network after filtering should be relevant to the products of interests, e.g., either containing the names of the products or the URL links to the product pages in the e-commerce site. Without loss of generality, we assume the customer-product network and the social network share the same sets of products of interests P. Here we focus on studying the social networks filtered with the product related terms.

Social Network: A social network is represented as $\mathcal{G}^s = (\mathcal{V}^s, \mathcal{E}^s)$, where $\mathcal{V}^s = \mathcal{U}^s \bigcup \mathcal{P}$ is the set of nodes including two types of nodes. $\mathcal{U}^s$ is the set of users and $\mathcal{P}$ is the set of the products of interests mentioned in the user-generated contents. $\mathcal{E}^s \subset \mathcal{V}^s \times \mathcal{V}^s$ denotes the set of edges in the network $\mathcal{G}^s$. The types of edges include the social links between users, the links between users and the products mentioned by the users, represented by $\mathcal{U}^s \times \mathcal{U}^s$ and $\mathcal{U}^s \times \mathcal{P}$, respectively.

Customer Identification: Suppose we have a customer-adoption network $\mathcal{G}^c$ and a social network $\mathcal{G}^s$, with a small set of identified pairs $\mathcal{A}$, the task of customer identification is to find the optimal set $\mathcal{A}^*$ in which all the customers in $\mathcal{G}^c$, who can be identified in $\mathcal{G}^s$, are matched to their corresponding accounts in $\mathcal{G}^s$.

Given a candidate pair (u_i^c, u_j^s) of a customer u_i^c in $\mathcal{U}^c$ and a social network user u_j^s in $\mathcal{U}^s$, we shall decide whether this pair belongs to the same individual. Let $f(u_i^c, u_j^s)$ denote the *customer identification function*, i.e.,

$$f(u_i^c, u_j^s) = \begin{cases} 1, & \text{if } u_i^c \in \mathcal{U}^c,\ u_j^s \in \mathcal{U}^s \text{ and} \\ & (u_i^c, u_j^s) \text{ belong to the same individual,} \\ 0, & \text{otherwise.} \end{cases}$$

Recall that each customer can only be identified as at most one (primary) user account in a social network and vise versa, i.e., one-to-one constraint. Hence, the set of known pairs $\mathcal{A}$ can be defined in the following formula:

$$\begin{aligned} \mathcal{A} = \{(u_i^c, u_j^s) | f(u_i^c, u_j^s) = 1, \text{and} \\ \nexists u_{i'}^c, u_{j'}^s, \text{s.t. } f(u_{i'}^c, u_j^s) = 1 \text{ or } f(u_i^c, u_{j'}^s) = 1\} \end{aligned}$$

, where $i \neq i'$ and $j \neq j'$. The optimum set $\mathcal{A}^*$ is the maximum set of $\mathcal{A}$, since $\mathcal{A}^*$ contains all the customers who can be identified in the social network. In addition, due to the one-to-one constraint, $\mathcal{A}^*$ is unique, i.e., no other combination of pairs that have the same size as $\mathcal{A}^*$.

The customer identification task serves as a prerequisite for developing many potential marketing applications in general e-commerce sites, as we have discussed in the Introduction. However, it involves two key challenges that make it difficult to be solved by applying existing social link prediction techniques (Getoor and Diehl, 2005; Liben-Nowell and Kleinberg, 2003;

Lichtenwalter et al., 2010; Kong et al., 2013b). First, we need to ensure one-to-one relationships between the target links to be predicted across networks with completely different schema (e.g., a customer-product network and a social network). To predict the existence of target links, we shall compare the similarity between pairs of nodes across networks. However, most existing features for link prediction, such as number of common neighbors, are designed for predicting the target links within a single network. The social features that exploit the social connections of identified pairs across networks are also not applicable, since there are no connections between customers in customer-product networks. How can we extract informative features for this customer identification task using basic information available in most e-commerce sites? Second, we should consider the prediction of all the target links collectively, not only because of the one-to-one constraint but, more importantly, because the nature of multiple networks tends to be partially aligned. How can we effectively match all the customers, who can be identified in social networks, to their corresponding social user accounts?

2.3 Methodology

In this section, we introduce a novel method, CSI (Customer-Social Identification), for effectively identifying customers in social networks. It consists of two phases, each of which addresses one major challenge of customer identification. The first phase tackles the feature extraction across networks with different schema, while the second phase manages to identify customers in partially aligned networks.

2.3.1 Extracting features across networks with different schema

As the first phase, CSI constructs the features that can be used to measure the similarity between pairs of accounts across networks with different schema. Because individuals often exhibit consistent behavioral patterns across networks, such as selecting similar usernames and passwords (Yan et al., 2000; Zafarani and Liu, 2013; Liu et al., 2013), we can make use of the similarities between candidate pairs to discover the same individuals.

Considering our purpose is to provide a general model for most e-commerce sites, we shall extract features by using the basic customer information which is generally available. Therefore, two common information sources are investigated: user profiles and the (product) interests of users. In the following, we present several similarity measures corresponding to each information source. The scores of these measures will be treated as the features for the next phase.

When a customer registers an account in an e-commerce site, s/he is usually asked to select a unique username and to fill in her/his full name and email address. This registration builds up the basic user profile of the customer. Other attributes, such as the city of residency, gender and ages, are also useful to identify individuals. Though, these attributes are inconsistent in multiple sites and often left blank by the customer. Hence, we attempt to measure the similarity mainly by exploiting names and email addresses.

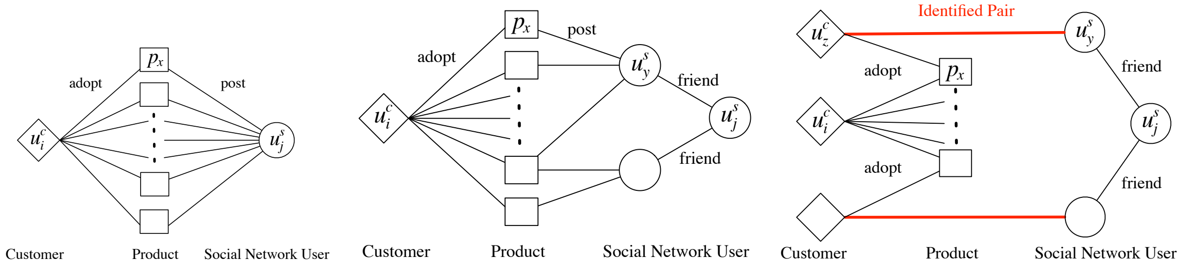
Names: Usernames are unique on each web site and can be viewed as identifiers of individuals, whereas the full names, i.e., the combinations of first name and last name, are not unique. (Zafarani and Liu, 2013) observed that human tends to have consistent behavior patterns when selecting usernames in different social media sites. For example, individuals often select new usernames by changing their previous usernames, such as add prefixes or suffixes or abbreviate part of their full names. However, their study mainly focus on the assumption that multiple prior usernames of the same individuals are available. This may not be an appropriate assumption in our problem, because most e-commerce sites usually obtain only one single prior username of each customer.

Therefore, among the top 10 important features presented in (Zafarani and Liu, 2013), we select the four features that can be calculated by the single prior username. Besides, we also consider the Levenshtein Edit Distance (Levenshtein, 1966), which can capture the changes of candidate usernames, as another feature. The five features are listed as follows:

- Exact username match,
- Jaccard similarity between the alphabet distribution of the candidate username and the prior username,
- Distance traveled when typing the candidate username using the QWERTY keyboard,
- Longest common subsequence between the candidate username and the prior username,
- Levenshtein edit distance.

Email: Email addresses can uniquely identify individuals, whereas they are not public available in most online social networks. In this chapter, email addresses are used as for verification of the identification. Once we discover that they exist in both customer profiles and user profiles in online social networks, we can pair the both accounts of their owners and put them into the set of identified pairs.

Modeling user interest similarity In additional, the products that adopted by customers and mentioned by social network users reflect their common interests to some extent. Therefore,



(a) common interests

(b) common interests with friends

(c) common interests with identified friends

Figure 2. Modeling user interest similarity

we propose to extract user interest features based on the similarity between the products of interests that customers and social network users both have in common.

1) **Common Interests (CI):** The most direct implementation of this idea for customer identification is to consider the number of interests that customer u_i^c and social network user u_j^s both have in common. We denote the interests of u_i^c as $\Gamma_p(u_i^c)$ and the interests of u_j^s as $\Gamma_p(u_j^s)$. The score of common interests is defined as follows:

$$\begin{aligned} score(u_i^c, u_j^s) &= |\{p_x|(u_i^c, p_x) \in \mathcal{E}^c\} \cap \{p_y|(u_j^s, p_y) \in \mathcal{E}^s\}| \\ &= |\Gamma_p(u_i^c) \cap \Gamma_p(u_j^s)| \end{aligned} \tag{2.1}$$

where $|\mathcal{P}|$ is the cardinality of the set $\mathcal{P}$.

2) **Jaccard's Coefficient (JC):** The Jaccard's coefficient is a normalized version of common interests, i.e., the number of common interests divided by the total number of distinct products of interests in $\Gamma_p(u_i^c) \cup \Gamma_p(u_j^s)$.

$$score(u_i^c, u_j^s) = \frac{|\Gamma_p(u_i^c) \cap \Gamma_p(u_j^s)|}{|\Gamma_p(u_i^c) \cup \Gamma_p(u_j^s)|} \tag{2.2}$$

3) **Admic/Adar Index (AA) (Adamic and Adar, 2003):** The AA index refines the simple counting of common interests by weighting rarer interests more heavily. We denote the customers who adopt p_x as $\Gamma_u^c(p_x)$ and the social network users who mention p_x as $\Gamma_u^s(p_x)$. We extend the AA index into multi-network settings, where the common interests are weighted by their average degrees in log scale. The similarity score of u_i^c and u_j^s is derived as follows:

$$score(u_i^c, u_j^s) = \sum_{\forall p_x \in \Gamma_p(u_i^c) \cap \Gamma_p(u_j^s)} log^{-1}(\frac{|\Gamma_u^c(p_x)| + |\Gamma_u^s(p_x)|}{2}) \tag{2.3}$$

4) **Resource Allocation Index (RA) (Zhou et al., 2009):** The RA index is similar to the AA index except the weight is distributed averagely instead of in log scale.

$$score(u_i^c, u_j^s) = \sum_{\forall p_x \in \Gamma_p(u_i^c) \cap \Gamma_p(u_j^s)} (\frac{|\Gamma_u^c(p_x)| + |\Gamma_u^s(p_x)|}{2})^{-1} \tag{2.4}$$

Above four measures compute the similarity between customer u_i^c and social network user u_j^s based on their shared (products of) interests directly, as illustrated in Figure 2(a). However, customer u_i^c may not actively mention the products that s/he has adopted in social networks.

To compute the interest similarity between u_i^c and u_j^s, we need to seek other connections or paths between them.

According to the researches of social influence on purchase behaviors (Hill et al., 2006; Guo et al., 2011; Bhatt et al., 2010), a customer is more likely to buy a product if his/her friends have widely adopt it. Thus, we consider utilizing the interests of friends to help locate the inactive customers. There are two types of paths between u_i^c and u_j^s through the interests of friends we can exploit. Figure 2(b) shows an example of the first type of a path. In Figure 2(b), the product p_x mentioned by u_y^s, a friend of u_j^s, is also adopted by u_i^c. If u_i^c and u_j^s belong to the same individual, this path $\langle u_i^c, p_x, u_y^s, u_j^s \rangle$ would imply the adoption of p_x is related to the post from u_y^s. The second type of a path is similar to the first one, except this time we will make use of the identified pairs. For example, in Figure 2(c), the product p_x adopted by u_z^c, who is identified as u_y^s (a friend of u_j^s), is also adopted by u_i^c. Similar to the first case, this path $\langle u_i^c, p_x, u_z^c(u_y^s), u_j^s \rangle$ also imply the adoption of p_x made by u_j^s is related to that made by $u_y^s(u_z^c)$, if u_i^c and u_j^s belong to the same individual.

Note that the common interests with (identified) friends is a weaker indicator than the common interests for a candidate pair. In this work, we extend the *Katz's* index (Katz and Katz, 1953) to provide a weighted measure on the collection of paths between u_i^c and u_y^s.

5) **Katz's Index (Katz and Katz, 1953):** The Katz's index sums over the collection of paths, which are exponentially damped by the length in order to count short paths more heavily, leading to the β-parameterized measure.

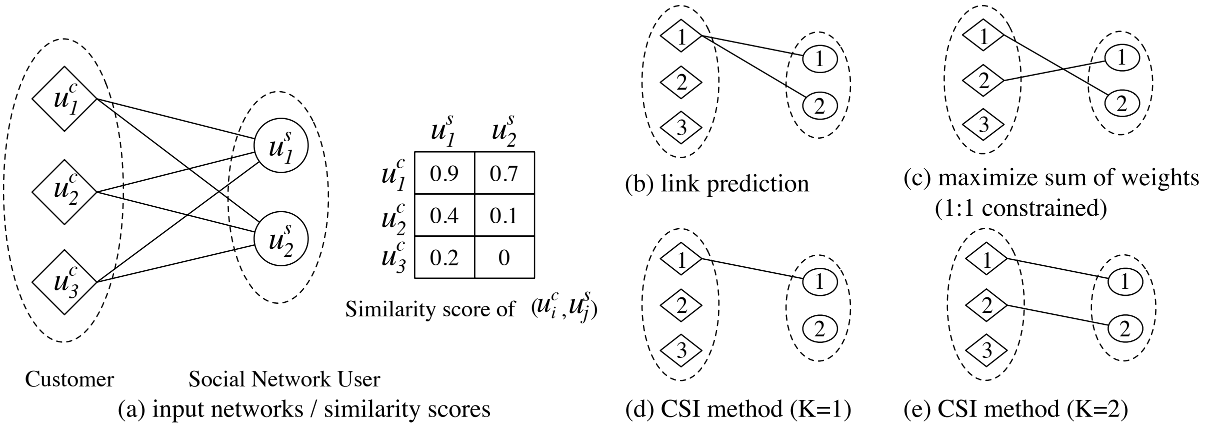
$$score(u_i^c, u_j^s) = \sum_{l=1}^{l_{max}} \beta^l \cdot |paths_{u_i^c,u_j^s}^{\langle l \rangle}| \tag{2.5}$$

where $paths_{u_i^c,u_j^s}^{\langle l \rangle}$ is the set of all length-l paths from u_i^c to u_j^s. Here we adapt the truncated Katz score, in which the length-l is limited to l_{max} instead of ∞ as in the original Katz's measure, since the truncated Katz often outperforms Katz for link prediction (Lu et al., 2010). In this work, we set $l_{max} = 2$ to capture both factors of the common interests and common interests with (identified) friends. $|paths_{u_i^c,u_j^s}^{\langle 1 \rangle}|$ is the same as the number of common interests, while $|paths_{u_i^c,u_j^s}^{\langle 2 \rangle}|$ is the number of paths through the interests of friends. For example, there are 5 paths between u_i^c and u_j^s in Figure 2(b) and 2 paths between them in Figure 2(c), and thus $|paths_{u_i^c,u_j^s}^{\langle 2 \rangle}| = 5 + 2 = 7$.

2.3.2 Connecting Users in partially aligned networks

With the features extracted in the previous phase, we can train a binary classifier (e.g., SVM or logistic regression) to roughly decide whether candidate pairs across networks belong to the same identities or not. However, the predictions of the binary classifier cannot be directly used for customer identification. This is because the inference of conventional classifiers are designed for constraint-free settings (e.g., one customers can be paired with multiple user accounts in a social network), and thus the one-to-one constraint on account pairs across networks may not hold.

Instead of simply relying on the decision made by the classifier, we notice that most classifiers also generate similarity scores for classification. Based on the similarity scores that are further calibrated (Zadrozny and Elkan, 2002), one may think of applying conventional match-



	u^s_1	u^s_2
u^c_1	0.9	0.7
u^c_2	0.4	0.1
u^c_3	0.2	0

(a) input networks / similarity scores

(b) link prediction

(c) maximize sum of weights (1:1 constrained)

(d) CSI method (K=1)

(e) CSI method (K=2)

Figure 3. Example of customer identification under different approaches. (a) is the input networks and similarity scores, (b) and (c) are the results of existing baselines. (d) and (e) are the results of CSI methods for $K = 1$ and $K = 2$, respectively.

ing techniques, such as stable marriage (Dubins and Freedman, 1981) and maximum weight matching, to find a one-to-one matching between pairs of accounts across two networks. Nevertheless, these techniques could be problematic in the customer identification task, since they usually assume networks are fully aligned, whereas in fact most networks are partially aligned. That is to say, some customers in an e-commerce site do not have any user accounts in an online social network. We should not pair these customers to any user accounts in the social network recklessly; otherwise, we may waste valuable resources on inappropriate targets.

In order to tackle the above issues, we propose to formulate the customer identification in partially aligned networks as a top-K maximum similarity and stable matching problem[1]. Specifically, our goal is to find the top K pairs that have the maximum similarity (or weight) among all the stable matching of any combination of K pairs across networks.

Generally speaking, stable matching is a one-to-one matching A with the principle that there is no unmatched pair (m, w) such that m and w both prefer each other to their current assignments in A. Here we say "m prefers w over z", if the pair score (m, w) is larger than the pair score (m, z). The primal reason for limiting our solution to stable matching is because stable matching methods can maximize the local benefits of one set of nodes. Other matching methods, such as *maximum weight matching*, are less suitable since they usually focus on maximizing the overall benefit of the mapping of the entire networks.

Take Figure 3 as an illustrative example of different methods. Suppose in Figure 3(a) we are given the similarity scores from the binary classifier for each candidate pair. Figure 3(b) shows that link prediction methods with a fixed threshold (e.g., 0.5) may not be able to predict well, because one customer could be linked with multiple accounts in the social network. On the other hand, "*maximum weight matching*" methods find a set of pairs with the maximum sum of weights (or similarities), in Figure 3(c), whereas it may not be a good solution for customer

[1] This problem is a variation of maximum weighted stable marriage (or royal couple matching in (Marie and Gal, 2007)) problem. The major difference is in that we aim at finding a one-to-one mapping for K nodes, instead of mapping all nodes.

identification. Since the similarity score of (u_1^c, u_1^s) is larger than that of (u_1^c, u_2^s), customer u_1^c is more likely to be the same individual of u_1^s rather than u_2^s.

Assuming K, the number of customers to be identified, is specified in advance, we propose to find the top K pairs of accounts with the maximum similarity, following the principle of stable matching mentioned above. Figure 3(d) shows an illustrative example of CSI with K=1. Pair (u_1^c, u_1^s) is the top-1 pair that has the maximum similarity score among all candidate pairs. Hence, we would identify u_1^s as the social network account of customer u_1^c. As a consequence, when K=2 in Figure 3(e), we should ignore the candidate pairs that associate with u_1^c or u_1^s due to the one-to-one constraint. Thus, the next pair we would choose is (u_2^c, u_2^s), whose score is the best among the rest pairs. In fact, among all the customers, probably only customer u_1^c has a user account, u_1^s, in the social network, because the scores of other customers do not indicate that they are similar enough to any users in the social network. Therefore, the result in Figure 3(d) is the most appropriate solution. Nonetheless, we should be able to find the top K-1 pairs before move to the K-th pair, which has lower similarity score than the top K-1 pairs.

The proposed CSI method for customer identification is shown in Algorithm 1. In each iteration, we select the pair of accounts (u_i^c, u_j^s) with the maximum similarity score from candidate pairs. If both u_i^c and u_j^s have not yet assigned to any account, we add (u_i^c, u_j^s) to the solution set $\mathcal{A}'$ and set u_i^c and u_j^s as occupied; otherwise if either u_i^c or u_j^s is occupied, we ignore (u_i^c, u_j^s). To facilitate the process of finding the pair with maximum score, we can maintain a max heap instead of a matrix to store the similarity scores of candidate pairs. The algorithm stops when

Algorithm 1 Customer-Social Identification

```
Input: A user-specified value $K$, a customer-product network $\mathcal{G}^c$, a social network $\mathcal{G}^s$,
    a set of existing identified pairs $\mathcal{A}$, .
Output: A set of predicted pairs $\mathcal{A}'$.
/* first phase*/
Construct a training set with known labels using $\mathcal{A}$
For each pair $(u_i^c, u_j^s)$, extract features
Train a classifier $C$ on the training set.
Inference using the trained $C$ on the test set.
/* second phase */
Calibrate the similarity scores of candidate pairs and sort them into a max heap $H$ by the scores.
Initialize all unlabeled $u_i^c$ in $\mathcal{G}^c$ and $u_j^s$ in $\mathcal{G}^s$ as free.
$\mathcal{A}' = \emptyset$
while $H \neq \emptyset$ and $|\mathcal{A}'| < K$ do
    Pop the pair $(u_i^c, u_j^s)$ with the max score from $H$
    if $u_i^c$ and $u_j^s$ are both free then
        $\mathcal{A}' = \mathcal{A}' \cup (u_i^c, u_j^s)$
        Set $u_i^c$ and $u_j^s$ as occupied
    end if
end while
```

the top K pairs are found, or there are no remaining candidate pairs in the max heap. The matching computed by the CSI method is guaranteed to be a stable matching, according to Theorem 1 in (Marie and Gal, 2007); furthermore, it has the maximum similarity score among all the stable matching of any combination of K pairs across networks, which can be easily proved by mathematical induction. Due to lack of space, we skip all the proofs.

It is worth noting that the selection of the parameter K is a challenging issue for most problems that need to find out the top-K elements. Different approaches are proposed for finding K, such as cross-validation and bootstrapping. In fact, the selection of K can also be implemented in other ways. For example, instead of setting K directly, one can find the top similar pairs until the similarity score of the matching pair is less than a threshold.

2.4 Experiments

In this section, we first introduce the data sets for the experiments, and then present experimental results as well as empirical analysis.

2.4.1 Data Preparation

We conduct the experiments on the real-world datasets spanning 10 months, as summarized in Table III. We choose Kickstarter.com, one of the largest sites for crowdfunding[1], as an e-commerce site because the adoption histories of each customer are public available. More importantly, novel and creative crowdfunding projects are notably discussed on Twitter where users are willing to share their interests.

Twitter: We gathered all the tweets regarding Kickstarter from Nov. 2012 to Sep. 2013. For each tweet's author, we queried Twitter API for the metadata about the author as well as the social links of the author. For each project in Kickstarter we consider only the tweets that can link to its webpage. We further filtered out the projects that were seldom discussed (less than 5 tweets) in Twitter. The Twitter dataset after filtering consists of 3,725 projects, 178K users, 5.4 million social links and 385K tweets that construct 234K links between Twitter users and projects.

Kickstarter: We recorded all the projects in Kickstarter launched after Nov. 2012 and completed before Sep. 2013. For each project, we obtained all of its backers, which can be viewed as its customers. For each customer, we crawled his/her user profile and recorded his/her

[1] Crowdfunding – in which people can propose projects and raise funds through collaborative contributions of crowd.

Twitter account, if available. The Kickstarter dataset after filtering consists of 3,725 projects, 545K customers and 868K adoption links between customers and projects. The detailed analysis of these datasets is available in (Lu et al., 2014b).

Data preprocessing: We preprocess the raw data to obtain the ground truth of identified pairs. If a customer, in the Kickstarter dataset, has shown his/her Twitter account in his/her user profile, and the Twitter account also exists in the Twitter dataset, we can safely treat the pair of accounts of the customer as an identified pair. The identified pairs represent the positive instances and they can be used to help construct negative instances of pairs. Due to the one-to-one constraint, we can easily find a negative pair by taking one account from an identified pair and connecting it to any account, in the opposite network, other than the corresponding one. Thus, we can obtain up to 1.3 billion negative pairs.

However, in practice, if an e-commerce company wants to identify one of its customer in a social network, it would probably inquire for the social network accounts whose usernames are similar to the names (i.e., username and full name) of the customer in the company. Consequently, it is critical for the e-commerce company to distinguish the actual one from others with very similar usernames. To simulate the query process, we shall select the negative pairs in which two accounts are likely to have similar usernames. Hence, for each account in the identified pairs, we search the candidate accounts, whose usernames contain a part of the names of the given account, in the opposite network. Then, the candidate accounts are ranked by the Levenshtein edit distance between the candidate usernames and the given customer username. Finally, we sample negative pairs by connecting the given account with up to 100 candidate

TABLE III

Statistics of the datasets.

Kickstarter-Twitter

	property	**original networks**	**sampled networks**
# node	projects	3,725	3,725
	customers	545,638	20,514
	social network users	178,792	43,675
# link	adoption	868,050	39,480
	post	234,550	58,988
	social links	5,467,565	513,651
	identified pairs	1,819	1,819
	negative pairs	1.3 billion	93,436

accounts, other than the corresponding one, with the smallest edit distance. The statistics of the original networks and the sampled networks are presented in Table III. In the following experiments, we mainly conducted on the sampled networks.

2.4.2 Comparative methods

We compare our CSI method with eight baselines, including both supervised and unsupervised link prediction methods, which are summarized as follows:

1) **Unsupervised Link Prediction methods:** We compare with a set of unsupervised link prediction methods using the user interest features discussed in Section 2.3.1: *Common Interests* (**CI**), *Jaccard Coefficient* (**JC**), *Adamic/ Adar* index (**AA**), *Resource Allocation* index (**RA**), and *Katz's* index (**Katz**). Following the setting in (Liben-Nowell and Kleinberg, 2003), we test the performance of Katz with three different values of β (i.e., 0.05, 0.005 and 0.0005).

Each link predictor outputs a ranked list of candidate pairs in deceasing order of similarity scores. We can evaluate the performance of an unsupervised method based on the ranked list.

2) **Supervised Link Prediction methods:** We test supervised link prediction methods using different types of feature sets separately. As discussed in Section 2.3.1, two feature sets are considered, i.e., **Profile** and **Interest**. We also compare with the combination of both sets of features (**Profile+Interest**). The label predictions of the base classifier are directly used as the final predictions.

3) **Customer-Social Identification (CSI):** The proposed method in this chapter. CSI leverages all the extracted features, i.e., Profile+Interest, for training the base classifier. Based on the scores generated by the classifier, CSI takes the top-K maximum similarity and stable matching as the final predictions. In default, K is set as the size of real identified pairs in the testing set. We will analyze the performance of CSI method with K varied in the experiment.

Evaluation Measures. We evaluate the performance of each method in terms of Precision, Recall, F1-score and area under ROC curve (AUC). The first three measures can evaluate the link prediction performances, while AUC evaluates the ranking performances. Since unsupervised methods only predict a real value for each candidate pair, we only compare the AUC of the unsupervised methods. Moreover, CSI and Profile+Interest share the same set of features and thus they have the same ranking scores generated by the base classifier. Hence, for AUC measure, we use CSI to represent both methods. For fair comparisons, linear LibSVM (Chang and Lin, 2011) is used as the base classifier for all the compared methods. Accuracy

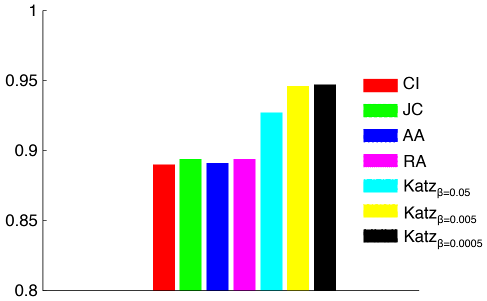


Figure 4. Comparison of customer identification using different features

is not included in the evaluation measures, since we mainly focus on the real-world imbalanced datasets in which Accuracy is usually meaningless.

Noteworthily, the F1-score and Recall of maximum weight matching (MWM) are consistently lower than 0.1, and the Precision and AUC of MWM are consistently lower than 0.2, which are significantly worse than those of other baseline methods. This is because MWM aims at maximizing the overall benefit of the entire matching instead of the local benefits of individuals, as mentioned in Section 2.3.2. Since MWM is not suitable for the customer identification problem, MWM is not listed as one of the competitive methods.

2.4.3 Performance Analysis

We conduct the experiments using 5-fold cross validation, where one fold is used as training data, and the remaining folds are used as testing data. We report the average results and standard deviations of 5-fold cross validation on the dataset.

We first investigate the performance of different features in the unsupervised methods. In Figure 4, Katz's methods outperform the methods using other features. It indicates that by exploiting the paths through the interests of friends, we have a better opportunity to identify customers in a social network. Even though the customers are not active in the social network, the common interests with friends may leak the information of the customers and direct us to identify them. However, from the comparison between Katz's methods with different β, which exponentially decreases the weight of longer path, we notice that the importance of friends' interest should not be overrated.

Next, the customer identification problem in real world involves distinguishing the real social network account of a customer from many other similar candidates. If we consider the real pair of accounts as a positive instance and other candidates as negative instances, the number of negative instances usually dominates that of positive instances. In other words, the data instances are usually imbalanced. It is crucial to identify customers in such imbalanced datasets.

Thus, we compare each method with imbalanced datasets by sampling pairs of accounts according to different imbalance ratios in each round of the cross validation. The imbalance ratio is defined as the number of negative pairs divides by the number of positive pairs. Table IV presents the performance of each method under different imbalance ratios. The best performances on each of the evaluation criteria are listed in bold. The results show that Profile features can be used as the most precise tool to identify some positive pairs but cannot cover most of them. By taking both Profile and Interest features into account, we are able to identify the majority of positive pairs effectively, while only slightly decreasing the precision of identification. The performance can be further improved through the one-to-one matching step in the

TABLE IV

Performance comparison with various imbalance ratios.

Measure	Methods	imbalance ratio				
		10	20	30	40	50
F1-score	Profile	0.672 ± 0.003	0.671 ± 0.003	0.666 ± 0.004	0.665 ± 0.003	0.661 ± 0.004
	Interest	0.836 ± 0.008	0.815 ± 0.011	0.78 ± 0.012	0.763 ± 0.01	0.747 ± 0.006
	Profile+Interest	0.895 ± 0.005	0.875 ± 0.014	0.847 ± 0.017	0.833 ± 0.019	0.803 ± 0.017
	CSI	**0.926± 0.004**	**0.915± 0.003**	**0.898± 0.006**	**0.89± 0.002**	**0.878±0.004**
Precision	Profile	**0.981± 0.001**	**0.977± 0.001**	**0.955± 0.004**	**0.952± 0.002**	**0.935± 0.004**
	Interest	0.944± 0.002	0.932±0.005	0.9±0.006	0.884±0.012	0.868±0.017
	Profile+Interest	0.959±0.004	0.941±0.005	0.925±0.008	0.911±0.006	0.896±0.008
	CSI	0.933± 0.003	0.92±0.003	0.902±0.006	0.894±0.003	0.881±0.003
Recall	Profile	0.511±0.004	0.511±0.004	0.511± 0.004	0.511±0.004	0.511±0.004
	Interest	0.75±0.014	0.725±0.019	0.688±0.021	0.671±0.017	0.656±0.012
	Profile+Interest	0.838±0.009	0.818±0.027	0.782±0.034	0.769±0.035	0.729±0.033
	CSI	**0.92±0.004**	**0.91±0.003**	**0.895±0.007**	**0.887±0.002**	**0.875±0.004**
AUC	Profile	0.791±0.001	0.791±0.001	0.791±0.001	0.792±0.001	0.792±0.001
	Interest	0.933±0.019	0.933±0.019	0.933±0.019	0.924±0.024	0.903±0.018
	CSI	**0.957±0.003**	**0.958±0.004**	**0.958±0.003**	**0.958±0.003**	**0.958±0.003**

proposed CSI method. As shown in Table IV, CSI consistently outperforms the other methods in F1-score, Recall and AUC with up to 38%, 80% and 21% improvement, respectively.

Another challenge of customer identification is that, in practice, there are only a small number of identified pairs. Hence, we next study the performance of each method using a small set of identified pairs for training. In each round of cross validation, we randomly sample a percentage of identified pairs from the training fold and use them for training. The results of all compared methods are reported in Table V. Again, CSI method consistently outperforms other methods in F1-score, Recall and AUC. Especially when only 20% of identified pairs from the training fold are used, the F1-score increases from 0.342 to 0.875 (with 156% improvement) and the Recall increases from 0.211 to 0.875 (with 315% improvement). We also notice that the performance of CSI method is quite stable with the change of the number of training samples.

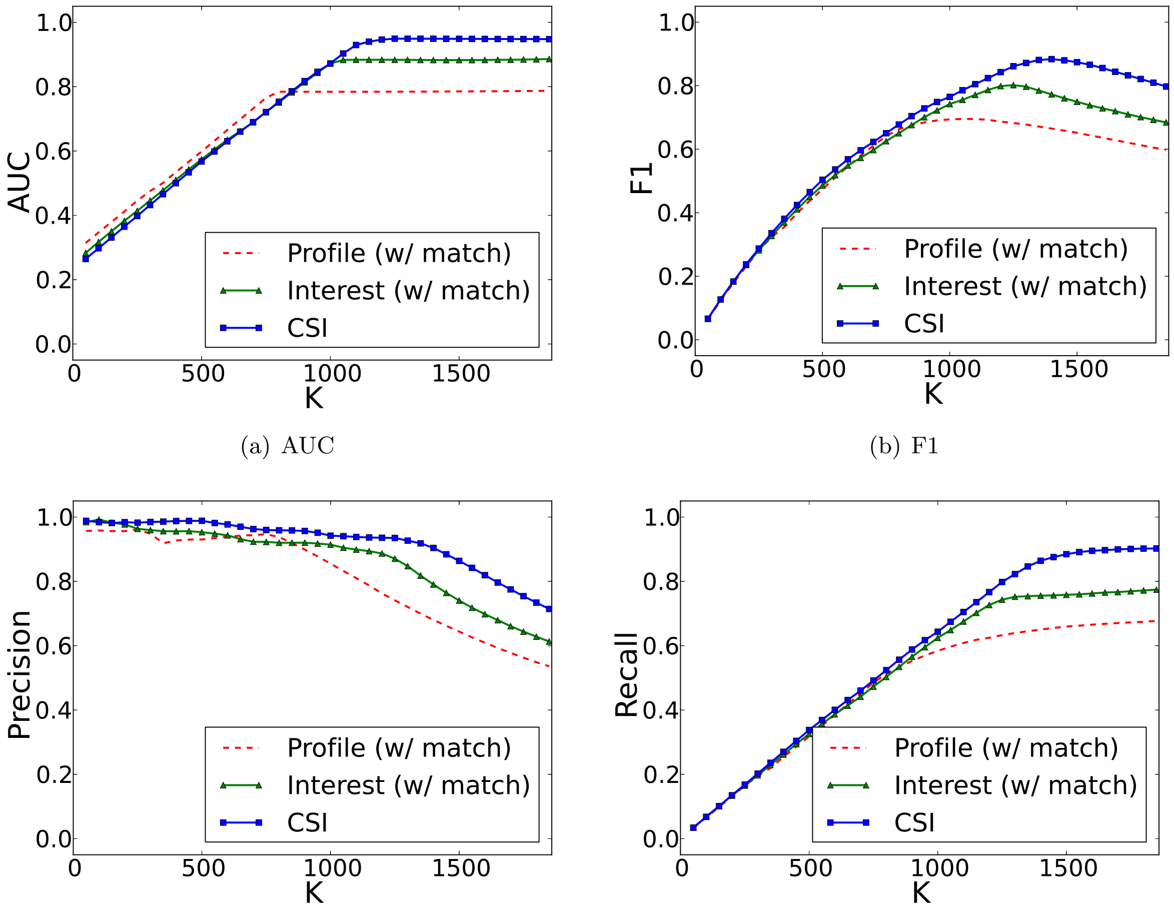
TABLE V

Performance comparison with various ratio of identified pairs.

Measure	Methods	number of identified pairs in training set (%)				
		20%	40%	60%	80%	100%
F1-score	Profile	0.342 ± 0.007	0.469 ± 0.159	0.661 ± 0.004	0.661 ± 0.003	0.661 ± 0.004
	Interest	0.486 ± 0.044	0.631 ± 0.026	0.695 ± 0.012	0.720 ± 0.022	0.747 ± 0.006
	Profile+Interest	0.620 ± 0.029	0.753 ± 0.007	0.771 ± 0.014	0.793 ± 0.013	0.803 ± 0.017
	CSI	**0.875± 0.005**	**0.878± 0.004**	**0.877± 0.006**	**0.878± 0.003**	**0.878±0.004**
Precision	Profile	**0.911± 0.006**	**0.924± 0.008**	**0.936± 0.002**	**0.936± 0.004**	**0.935± 0.004**
	Interest	0.900± 0.030	0.893±0.008	0.892±0.003	0.884±0.006	0.868±0.017
	Profile+Interest	0.938±0.016	0.914±0.015	0.900±0.006	0.901±0.005	0.896±0.008
	CSI	0.875± 0.005	0.877±0.004	0.876±0.006	0.878±0.003	0.881±0.003
Recall	Profile	0.211±0.006	0.331±0.150	0.511± 0.004	0.511±0.004	0.511±0.004
	Interest	0.335±0.041	0.489±0.032	0.569±0.016	0.609±0.034	0.656±0.012
	Profile+Interest	0.464±0.033	0.641±0.008	0.674±0.022	0.708±0.021	0.729±0.033
	CSI	**0.875±0.005**	**0.878±0.004**	**0.877±0.006**	**0.879±0.003**	**0.875±0.004**
AUC	Profile	0.773±0.025	0.793±0.002	0.794±0.002	0.793±0.002	0.792±0.001
	Interest	0.903±0.018	0.903±0.018	0.894±0.002	0.904±0.020	0.903±0.018
	CSI	**0.958±0.003**	**0.958±0.003**	**0.958±0.004**	**0.958±0.004**	**0.958±0.003**

This is because CSI method is designed to find the best stable matching all the time. Lack of training samples only affect the accuracy of similarity scores, while it probably would not change the preference of each account.

Finally, we investigate the performance of each method with K varied, where K is the number of pairs we should find in a one-to-one matching. In our experiments, K is set as 1466, which is the size of real pairs in our testing set, in default. Since the predictions of classifications cannot be directly compared, we won't be able to find the top-K pairs using the above baseline methods. Thus, in this experiment, we compare the performance of the CSI methods using different sets of features. We denote the CSI method using only Profile feature set as "Profile (w/ match)", and we denote that using only Interest feature set as "Interest (w/ match)". Figure 5 shows that CSI method incorporating the more features can achieve



(a) AUC

(b) F1

(c) Precision

(d) Recall

Figure 5. Comparison between CSI and baselines with K varied

the better performance. Besides, the CSI method using only the Interest feature set performs better than that using only the Profile feature set. More importantly, our proposed CSI method achieves the best performance when K is around 1466, the actual size of pairs to be identified. This indicates that CSI method can effectively find the top-K pairs that are most likely to be the real pairs before it moves to pick the less possible pairs.

2.5 Related Work

Due to the emergence of online social network services, social network analysis have been intensively studied in recent years (Lu et al., 2010; Kong et al., 2013b; Lichtenwalter et al., 2010; Getoor and Diehl, 2005). One active research topic is to predict unknown link in social network. (Liben-Nowell and Kleinberg, 2003) developed unsupervised link prediction methods based upon several topological features. These proposed features can be further used for link prediction. Recently, there there many works on link prediction in social networks. For example, (Backstrom and Leskovec, 2011) proposed a supervised random walk algorithm to estimate the strength of link in social networks. (Lichtenwalter et al., 2010) discussed different challenges of link prediction. (Kong et al., 2013b) formulated the problem of connecting accounts across social networks as a anchor link prediction task, w.r.t one-to-one constraint across social networks. They leverage the heterogeneous features, such as social, spatial and temporal information, to help predict the anchor links.

Recently, user identification across multiple social networks has attracted many attentions (Zafarani and Liu, 2013; Liu et al., 2013; Raad et al., 2010; Malhotra et al., 2012). (Zafarani and Liu, 2013) observed that individuals often exhibit consistent behavioral patterns across networks when selecting usernames. Based on the observation, they proposed a behavior model to determine whether two usernames are belong to the same individual. (Raad et al., 2010) addressed the problem of matching user profiles for inter-social networks. (Malhotra et al., 2012) analyzed users' online digital footprints and applied context specific techniques to measure the similarity of accounts across networks. These studies indicate that username is one of the most discriminative features for disambiguating user profiles. However, customer identification has some unique properties that make it different to the previous works. First, it requires to predict links across networks with completely different schema (i.e., bipartite network vs. general heterogeneous network). Second, since most networks are partially aligned, we should identify the most similar pairs instead of mapping the entire networks. Due to these issues, previous approaches may not be directly applicable to customer identification.

CHAPTER 3

TRANSFER LEARNING FOR NEW DOMAINS

(This chapter was previously published as “Item Recommendation for Emerging Online Businesses”, in IJCAI ’16 (Lu et al., 2016).)

3.1 Introduction

Recommending products, services, friends etc. to users on social networks and e-commerce websites, is an important component of online businesses like LinkedIn, Amazon, Coursera, etc. Not only users can be guided to discover useful or interesting items, but a large amount of revenues can be generated by recommending the right products to the target users. To make accurate recommendation, one of the major tasks is to predict users’ ratings on items. Among many recommendation techniques, collaborative filtering (CF) methods (Koren, 2008; Rong et al., 2014) have been widely used in many areas, such as social networks and e-commerce sites. Finding similar users and items for recommendation based on historical user-item interactions, CF methods have proven to be one of the most successful solutions for recommendation in developed online businesses.

However, it is a different story when employing collaborative information for emerging online businesses. Generally speaking, emerging businesses can be online services that are still in its embryonic stage; or mature ones that start to branch into new geographic areas or new categories (Zhang and Yu, 2015). In an emerging business, available user data, including either explicit feedbacks (e.g., ratings) or implicit feedbacks (e.g., clicks), are usually too sparse for effective recommendation. Furthermore, it still remains a challenge to provide an accurate prediction for new users with extremely few records, which is called the "cold-start" problem.

Fortunately, different online businesses may share some common users and items. Figure 6 shows an example of an emerging e-commerce site and a developed review site. Note that among four customers in the e-commerce site, two of them also have user accounts in the review site. Besides, two products are shared by both sites. The recommendation task in this chapter is to predict the rating of a given item by a target customer, as shown in question marks at the bottom of Figure 6. Although user data are extremely sparse in the e-commerce site, abundant knowledge in the more developed review site can be utilized to help recommendation.

Recently, researchers have applied *transfer learning* to CF methods for alleviating the data sparsity in recommender systems (Singh and Gordon, 2008; Luo et al., 2014). Transfer learning is a technique to utilize auxiliary information sources to help the learning task in the target domain. Existing transfer learning models in CF typically utilize information shared by related domains to learn latent factors for better recommendations, where the latent factors are encoded into the low-rank representations (of a rating matrix) that can minimize the reconstruction error. Most of the models assume that there are correspondences between users or items across domains, and use the correspondences as constraint to align the latent factors. For example, collective matrix factorization (CMF) (Singh and Gordon, 2008) finds joint low-rank representations by simultaneously factorizing several matrices, sharing parameters among latent factors when an entity participates in multiple relations.

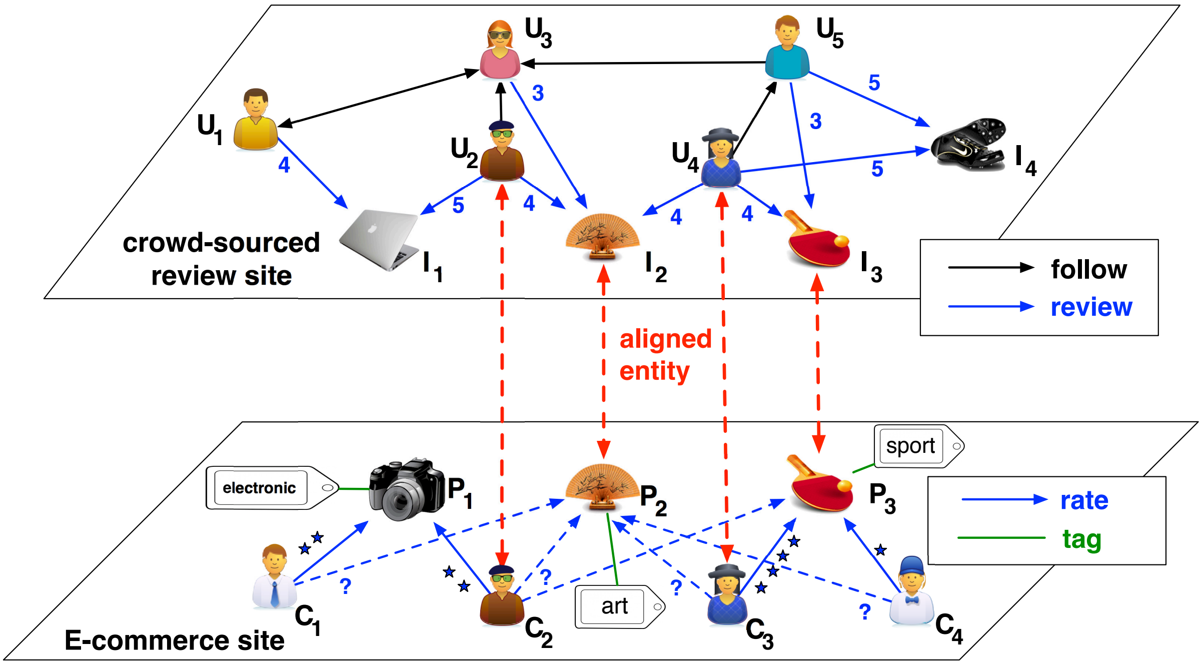


Figure 6. Example of paritally aligned heterogeneous networks, where some entities in one network belong to the same identities in the other network.

However, the prerequisite of entity correspondence across different domains is often hard to satisfy for all entities in most real-world scenarios. For example, only a portion of users and items are overlapped in both eBay and Epinion. Manually identifying the entity correspondences is expensive and time-consuming as users may use different names, or an item may be named differently in different online businesses. Different identification algorithms (Kong et al., 2013b; Zafarani and Liu, 2013; Lu et al., 2014a; Zhang et al., 2015) are designed to automatically mapping entities across domains but they are usually much less accurate. Further, the overlapping users can be biased to more active users. The others can only benefit from the transferred knowledge through the sparse user-item interactions in the target domain, leading to sub-optimal results.

Recent studies suggest that preserving geometric closeness between similar entities is critical to make transfer learning models robust (Zhu and Lafferty, 2005; Long et al., 2014). Specifically, to learn latent factors for recommendation in the target domain via transfer learning, both the cross-domain entity correspondences and the similarity information within the target domain should be jointly taken into consideration. Recall the underlying assumption of CF: similar users shall rate items similarly and similar items shall receive similar ratings from similar users. The entities with cross-domain correspondences are assumed to behave similarly across different domains, and thus their latent factors are aligned by CMF based methods. In the same manner, if we are able to find similar entities within target domain, we can refine the latent factors for similar entities such that the learned latent factors will preserve the similarity information of the entities. If we can derive such similarity information for all entities universally, non-overlapping entities can also benefit from the transferred knowledge through the latent factor refinement. Thus, the learned latent factors should be less biased to the overlapping portion.

With the information sparsity of the emerging businesses, though, it is non-trivial to derive entity similarity in emerging businesses using traditional similarity measures such as cosine similarity and Euclidean distance. A number of studies leverage linkage structure for measuring similarity (e.g., *SimRank* (Jeh and Widom, 2002) and *PathSim* (Sun et al., 2011)). These linkage based similarity measures typically treat the relationships between entities as binary connections (or with connection probabilities). However, relationships between entities are usually attached with plenty of attributes, e.g., the rating of an item given by a user, the timestamp of a post written by a user and the distance between two locations. Discarding these relationship attributes can lead to a degraded performance. How can we define an accurate similarity measure by keeping relationship attributes and further leveraging side information from multiple businesses?

In this chapter we introduce the concept of augmented meta path (AMP), which is a sequence of relations between entities and the relations are augmented with link attributes. Taking both the linkage structures and the augmented link attributes into account, we define a novel similarity measure called *AmpSim* that can judiciously capture the rich similarity semantics between entities via AMPs. By traversing between networks through overlapping entities, AmpSim can easily gather side information from other networks to help measure entity similarity. For example, the similarity between customer C_2 and C_3 in Figure 6 can be measured through the AMP instance $C_2 \leftrightarrow U_2 \xleftarrow{\text{[score=4]}} I_2 \xrightarrow{\text{[score=4]}} U_4 \leftrightarrow C_3$, even there is no connection between C_2 and C_3 in the emerging e-commerce site.

We further integrate the similarity information captured by AmpSim with a CMF model such that the latent factors of similar entities would be refined w.r.t the geometric structure. As a consequence, non-overlapping entities can also benefit from the transferred knowledge through the latent factor refinement. Hence, the transferred knowledge would not be biased to the more active ones but can be iteratively propagated across networks to fit the emerging business. Through extensive experiments on real-world datasets, the proposed model is demonstrated to significantly outperform other state-of-the-art CF algorithms in addressing item recommendation for emerging businesses.

3.2 Preliminaries

In this section, we present the preliminaries and the problem formulation of this study. Table VI lists the main notations we use through this chapter.

We extend the definition of Heterogenous Information Network (Sun et al., 2011) to take the link attributes augmented with the links into account.

Definition 1. Augmented Heterogenous Information Network (AHIN): An AHIN can be represented as $G = (\mathcal{V}, \mathcal{E}, \mathcal{Y})$, where each entity $v \in \mathcal{V}$ belongs to an entity type $\tau(v) \in T$, and each link $l \in \mathcal{E}$ belongs to a relation type $\psi(l) \in R$ and may have an augmented attribute y, which belongs to an link attribute type $\omega(y) \in \mathcal{W}$.

An online business can be modeled using an AHIN: $G = (\mathcal{V}, \mathcal{E}, \mathcal{Y})$, where the user set $\mathcal{U}$ and item set $\mathcal{I}$ are subsets of entities (i.e., $\mathcal{U}, \mathcal{I} \subset \mathcal{V}$), the existing feedbacks of items given by users are subsets of links (i.e., $\mathcal{U} \times \mathcal{I} \subset \mathcal{E}$). The attributes augmented with the feedbacks (such as the user-item ratings $\mathbf{Y}$, where each entry $\mathbf{Y}_{ui}$ corresponds to the rating of user $u \in \mathcal{U}$ on item $i \in \mathcal{I}$) are kept as link attributes in G (i.e., $\mathbf{Y} \subset \mathcal{Y}$). An emerging business is a business in which the average number of ratings is lower than a threshold, i.e., $AvgDeg(\mathbf{Y}) = \frac{|\mathbf{Y}|}{|\mathcal{U}|} < \epsilon$.

If pairs of different networks share some common entities, then these networks are called aligned networks.

Definition 2. Aligned Networks: Aligned networks can be formulated as $\overline{G} = ((G^{(1)}, G^{(2)}, \ldots, G^{(\Pi)}), \mathcal{A})$, where $\mathcal{A} = \bigcup_{s,t} \mathcal{A}^{(s,t)}, 1 \leq s < t \leq \Pi$. $\mathcal{A}^{(s,t)}$ is the set of undirected anchor links $(v_p^{(s)}, v_q^{(t)})$ between entities across network $G^{(s)}$ and $G^{(t)}$, where $v_p^{(s)} \in \mathcal{V}^{(s)}$ and $v_q^{(t)} \in \mathcal{V}^{(t)}$.

We can depict the network schema of an AHIN G as $S_G = (T, R, W)$, and the network schema of of an aligned networks $\overline{G}$ as $S_{\overline{G}} = (\overline{T}, \overline{R}, \overline{W})$, where $\overline{T} = \bigcup_\pi T^{(\pi)}$ is the union of entity types and $\overline{R} = \bigcup_\pi R^{(\pi)} \bigcup R^A$ is the union of link types (including the anchor link relation R^A), and $\overline{W} = \bigcup_\pi W^{(\pi)}$ is the union of link attribute types, respectively. The network schema of the aligned networks in Figure 6, for instance, is shown in Figure 7, where rectangles, diamonds and circles represent the entity types, relation types and link attribute types, respectively.

For the sake of simplicity, we focus on one target emerging business network $G^{(t)}$ and one source network $G^{(s)}$ (i.e., $\Pi = \{s, t\}$), which can be easily extended to multiple related business. The task in this chapter is to predict the missing rating in the target business $G^{(t)}$. The predicted rating of user u on item j is denoted by $\hat{\mathbf{Y}}_{uj}^{(t)}$. Given aligned networks $\overline{G}$, we aims at providing a predictive function $f : \overline{G} \rightarrow \hat{\mathbf{Y}}^{(t)}$, such that the prediction errors are minimized.

3.3 Augmented Meta Path-based Similarity

To measure entity similarity by taking both the linkage structure and link attributes into account, we define the augmented meta paths as follows:

Definition 3. Augmented meta path (AMP): Given the network schema $S_G = (T, R, W)$, an AMP $\mathcal{P}$ is a sequence of relations augmented with link attributes between entities, and is denoted in the form of $T_1 \xrightarrow{R_1[W_1]} T_2 \xrightarrow{R_2[W_2]} \ldots \xrightarrow{R_{l-1}[W_{l-1}]} T_l$, where R_i is the relation type between entity type T_i and T_{i+1}, and W_i is the augmented link attribute type of R_i if exists.

TABLE VI

Notation summary

Symbol	Description
u, i	user, item
$\mathbf{Y}, \mathbf{I}$	user-item rating and indicator matrices
$\mathbf{P}, \mathbf{Q}$	low rank representations of users and items
G, S_G	augmented information network and its schema
$\mathcal{V}, \mathcal{E}, \mathcal{W}$	set of entities, links and link attributes
$\mathcal{A}^{(s,t)}$	set of anchor links across network $G^{(s)}$ and $G^{(t)}$
T, R	entity type and link type
$\overline{G}$	aligned networks
$\mathcal{P}, \mathcal{P}^{-1}$	meta-path and reverse meta-path

For simplicity, we will use the first letter of each entity to represent that entity, the opposite direction of the arrow to represent the reverse relation, ignore the relation type if there is no ambiguity, and ignore the link attribute type if it is simply the count of connections. If an AMP $\mathcal{P}$ does not involve any anchor link relations R^A, we call it as an intra-network AMP. Otherwise, we call it as an inter-network AMP.

Example 1. *Intra-network AMP:* The products rated by the same customer can be captured by $P \xleftarrow{\text{[star]}} C \xrightarrow{\text{[star]}} P$.

Example 2. *Inter-network AMP:* The products in $G^{(t)}$ that are reviewed by the same users in $G^{(s)}$ can be observed by $P \leftrightarrow I \xleftarrow{\text{[score]}} U \xrightarrow{\text{[score]}} I \leftrightarrow P$, where $\leftrightarrow$ denotes the anchor links across $G^{(s)}$ and $G^{(t)}$.

Inspired by the Homophily Principle (McPherson et al., 2001) – two entities are considered to be similar if they are related to similar neighbors – we propose to measure the similarity of

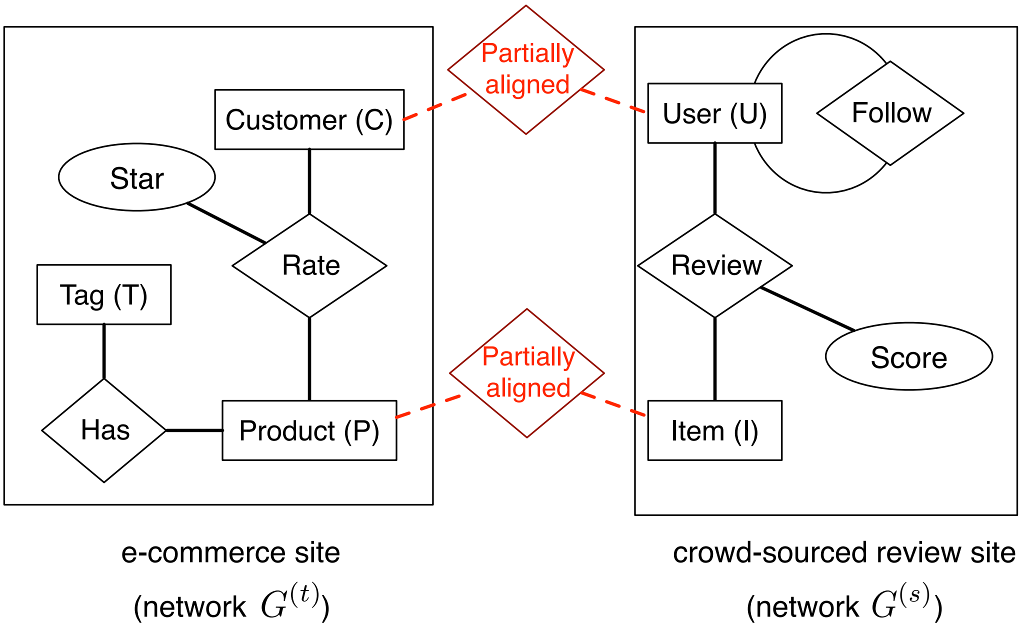


Figure 7. Schema of an e-commerce site partially aligned with a crowd-sourced review site.

two entities by their neighbors' similarities. However, an AMP may consist of multiple types of relations, and different link attributes are hard to be compared. To provide a general similarity measure, we apply a simple trick to normalize the link attributes in the AMP.

Let first assume that in the AMP the link attributes appear in pairs (i.e., the number of the same type of link attributes is even). We can replace the link attribute $\mathbf{Y}$ by the normalized value $\mathbf{M}$ using following equation:

$$\mathbf{M}_{ui} = \frac{Y_{ui} - b_i}{\sqrt{\sum_j (Y_{uj} - b_j)^2}} \tag{3.1}$$

, where b_i is a bias term for the entity i. For the link attributes that are bounded within a certain range (e.g., a rating can be bounded between 1 and 5), we set $b_i = \overline{Y_i}$ the average value of entity i. For the other type of attributes, we set $b_i = 0$. It is not hard to see that the multiplication of a pair of normalized link attributes (i.e., $\mathbf{M}\mathbf{M}^T$) equals to adjusted cosine similarity if $b_i \neq 0$ and equals to cosine similarity if $b_i = 0$. For the link attributes that do not appear in pairs, there is no way to compare the link attributes with others. Thus, we replace these attributes by the count of the connections and normalize them using Equation 3.1 with $b_i = 0$.

Considering the normalized link attributes and the direction of AMPs, we formulate the similarity measure as:

Definition 4. **AmpSim:** An augmented meta path-based similarity measure between v_a and v_b based on path $\mathcal{P}$ is:

$$s(v_a, v_b|\mathcal{P}) = \frac{[\prod_{i=1}^{l} \mathbf{M}_i]_{ab} + [\prod_{i=l}^{1} \mathbf{M}_i^T]_{ba}}{[\prod_{i=1}^{l} \mathbf{M}_i]_{a*} + [\prod_{i=l}^{1} \mathbf{M}_i^T]_{b*}} \in [0, 1],$$

where $\prod_{i=1}^{l} \mathbf{M}_i$ denotes the product of the normalized link attributes upon $\mathcal{P}$, $[M]_{ab}$ means the entry (a, b) in $\mathbf{M}$, and $[M]_{a*}$ means the ath row in $\mathbf{M}$. Since *AmpSim* can be computed in matrix form, the time complexity of computing AmpSim equals to that of matrix multiplications. Note that when $\mathcal{P}$ is symmetric and all the link attributes are the count of connections, *AmpSim* is equivalent to *PathSim* (Sun et al., 2011). Hence, AmpSim can be seen as a generalized version of *PathSim* on AMPs.

Different AMPs capture the similarity between entities in different aspects and overall similarity between entities can be obtained by aggregating information from all these AMPs. Without loss of generality, we choose *tanh* as the aggregation function, the overall similarity between entity v_a and v_b can be represented as

$$S(v_a, v_b) = \frac{tanh(\sum_i w_i S_i(v_a, v_b))}{tanh(1)} \in [0, 1], \tag{3.2}$$

where S_i denote the *AmpSim* upon the AMP $\mathcal{P}_i$, the value of w_i denotes the weight of $\mathcal{P}_i$, and $\sum_i w_i = 1$.

3.4 Recommendation Model

In transfer learning based collaborative filtering, collective matrix factorization (CMF) (Singh and Gordon, 2008) is proposed to estimate the low-rank representations as follows:

$$\min_{\mathbf{P}^{(\pi)}, \mathbf{Q}^{(\pi)}, \pi \in \{s,t\}} \sum_{\pi \in \{s,t\}} ||\mathbf{I}^{(\pi)} \odot (\mathbf{Y}^{(\pi)} - \mathbf{P}^{(\pi)} \mathbf{Q}^{(\pi)^T})||_F^2,$$

where $\odot$ is the Hadamard (element-wise) product, $|| * ||_F^2$ stands for Frobenius norm, and $\mathbf{I}^{(\pi)}$ is an indicator matrix. $\mathbf{I}_{ui}^{(\pi)} = 1$ if $\mathbf{Y}_{ui}^{(\pi)}$ is observed, and otherwise $\mathbf{I}_{ui}^{(\pi)} = 0$. $\mathbf{P}^{(\pi)} = [\mathbf{p}_1^{(\pi)}; \mathbf{p}_2^{(\pi)}; \ldots; \mathbf{p}_{n_\pi}^{(\pi)}] \in \mathbb{R}^{n_\pi \times k}$ and $\mathbf{Q}^{(\pi)} = [\mathbf{q}_1^{(\pi)}; \mathbf{q}_2^{(\pi)}; \ldots; \mathbf{q}_{m_\pi}^{(\pi)}] \in \mathbb{R}^{m_\pi \times k}$ are low-rank representation of users and items. n_π and m_π are the number of users and items in network $G^{(\pi)}$, and k is the parameter that estimate the rank. The key idea of CMF is to share parameters among factors when an entity participates in multiple relations. In (Singh and Gordon, 2008), for instance, the factor matrices of items are assumed to be the same (i.e., $\mathbf{Q}^{(s)} = \mathbf{Q}^{(t)}$).

We formulate our model as a constrained collective matrix factorization, where three soft constraints are involved:

- **Non-negativity**: The factor matrices $\{\mathbf{P}^{(\pi)}\}, \{\mathbf{Q}^{(\pi)}\}$ contain only nonnegative entries, since we only focus on positive interactions between users and items.
- **Geometric closeness**: The latent factors of similar entities should be close w.r.t the geometric structure. Preserving the geometric structure in the target domain can be achieved by the geometric regularization (Cai et al., 2011):

$$\mathcal{R}_G(\mathbf{P}) = \frac{1}{2}\sum_{u,v} \mathbf{S}_{uv}||\mathbf{p}_u - \mathbf{p}_v||_2^2,$$

  where $\mathbf{S}$ is computed by Equation 3.2.
- **Alignment constraint**: The latent factors of aligned entities should be close. Let (i, g_i) be the anchor link between entity i in the target network $G^{(t)}$ and entity g_i in the source network $G^{(s)}$. The difference of their latent factors should be minimized:

$$\mathcal{R}_A(\mathbf{Q}^{(t)}, \mathbf{Q}^{(s)}) = \frac{1}{2}\sum_{(i,g_i)\in\mathcal{A}^{(s,t)}} ||\mathbf{q}_i^{(t)} - \mathbf{q}_{g_i}^{(s)}||_2^2.$$

Integrating the alignment regularization $\mathcal{R}_A$ and the geometric regularization $\mathcal{R}_G$ seamlessly would enjoy the intrinsic mutual reinforcement learning: 1) with $\mathcal{R}_A$, knowledge can be transferred between businesses through the common latent factors of overlapping entities; 2) with

$\mathcal{R}_G$, the latent factors can be refined for similar entities to fit the geometric closeness within the emerging business. The objective function of our model can be formulated as follows:

$$\begin{aligned}\min_{\substack{\mathbf{P}^{(\pi)},\mathbf{Q}^{(\pi)}\geq 0,\\ \pi\in\{s,t\}}} \mathcal{J} = & \sum_{\pi\in\{s,t\}} ||\mathbf{I}^{(\pi)} \odot (\mathbf{Y}^{(\pi)} - \mathbf{P}^{(\pi)}\mathbf{Q}^{(\pi)^T})||_F^2 \\ & + \frac{\alpha}{2} \sum_{\pi\in\{s,t\}} (||\mathbf{P}^{(\pi)}||_F^2 + ||\mathbf{Q}^{(\pi)}||_F^2) \\ & + \beta(\mathcal{R}_A(\mathbf{P}^{(t)}, \mathbf{P}^{(s)}) + \mathcal{R}_A(\mathbf{Q}^{(t)}, \mathbf{Q}^{(s)})) \\ & + \lambda(\mathcal{R}_G(\mathbf{P}^{(t)}) + \mathcal{R}_G(\mathbf{Q}^{(t)})), \end{aligned} \tag{3.3}$$

where α controls the regularization to avoid over-fitting when learning $\{\mathbf{P}^{(\pi)}\}$, $\{\mathbf{Q}^{(\pi)}\}$, β controls the trade-off between source network and target network and λ controls the importance of geometric closeness. Since we focus on improving recommendation in the target business, the geometric regularization $\mathcal{R}_G$ is only applied to $\mathbf{P}^{(t)}$ and $\mathbf{Q}^{(t)}$.

To ease the subsequent derivation, we rewrite the geometric terms into trace form. From the similarity matrix $\mathbf{S}$ on users, we define the diagonal matrix $\mathbf{D}$ whose elements $\mathbf{D}_{ii} = \sum_j \mathbf{S}_{ij}$, and the Laplacian matrix $\mathbf{L}_P = \mathbf{D} - \mathbf{S}$. Then $\mathcal{R}_G(P)$ can be reduced into the trace form:

$$\sum_{u,v} \mathbf{S}_{uv}||\mathbf{p}_u - \mathbf{p}_v||^2 = Tr(\mathbf{P}^T(\mathbf{D} - \mathbf{S})\mathbf{P}) = Tr(\mathbf{P}^T\mathbf{L}_P\mathbf{P}).$$

Similar, given the similarity matrix $\mathbf{S}'$ on items, $\mathcal{R}_G(Q)$ can also be reduced into the trace form $Tr(\mathbf{Q}^T\mathbf{L}_Q\mathbf{Q})$, where $\mathbf{L}_Q = \mathbf{D}' - \mathbf{S}'$ and $\mathbf{D}'_{ii} = \sum_j \mathbf{S}'_{ij}$.

For solving Equation 3.3, we derive the multiplicative updating rules w.r.t $\{\mathbf{P}^{(\pi)}\}, \{\mathbf{Q}^{(\pi)}\}$ in the rest of this section.

Let $\mathbf{A}$ be the user mapping matrix that map users from network $\mathcal{G}^{(t)}$ to network $\mathcal{G}^{(s)}$, i.e., $\mathbf{A}_{u,g_u} = 1$ if $(u, g_u) \in \mathcal{A}^{(t,s)}$, zero otherwise. The partial derivatives of $\mathcal{J}$ in Equation 3.3 w.r.t $\{P^{(\pi)}\}$ are:

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \mathbf{P}^{(s)}} &= -(\mathbf{I}^{(s)} \odot \mathbf{Y}^{(s)})\mathbf{Q}^{(s)} + \mathbf{I}^{(s)} \odot (\mathbf{P}^{(s)}\mathbf{Q}^{(s)^T})\mathbf{Q}^{(s)} \\ &+ \alpha\mathbf{P}^{(s)} + \beta(\mathbf{A}^T\mathbf{A}\mathbf{P}^{(s)} - \mathbf{A}^T\mathbf{P}^{(t)}) \\ \frac{\partial \mathcal{J}}{\partial \mathbf{P}^{(t)}} &= -(\mathbf{I}^{(t)} \odot \mathbf{Y}^{(t)})\mathbf{Q}^{(t)} + \mathbf{I}^{(t)} \odot (\mathbf{P}^{(t)}\mathbf{Q}^{(t)^T})\mathbf{Q}^{(t)} \\ &+ \alpha\mathbf{P}^{(t)} + \beta(\mathbf{A}\mathbf{A}^T\mathbf{P}^{(t)} - \mathbf{A}\mathbf{P}^{(s)}) + \lambda\mathbf{L}_P\mathbf{P}^{(t)}\end{aligned}$$

Using the Karush-Kuhn-Tucker (KKT) complementarity conditions, we can obtain the following updating rules:

$$\begin{aligned}&\mathbf{P}^{(s)} = \mathbf{P}^{(s)} \odot \\ &\quad \sqrt{\frac{(\mathbf{I}^{(s)} \odot \mathbf{Y}^{(s)})\mathbf{Q}^{(s)} + \beta\mathbf{A}^T\mathbf{P}^{(t)}}{\mathbf{I}^{(s)} \odot (\mathbf{P}^{(s)}\mathbf{Q}^{(s)^T})\mathbf{Q}^{(s)} + \alpha\mathbf{P}^{(s)} + \beta\mathbf{A}^T\mathbf{A}\mathbf{P}^{(s)}}}, \\ &\mathbf{P}^{(t)} = \mathbf{P}^{(t)} \odot \\ &\quad \sqrt{\frac{(\mathbf{I}^{(t)} \odot \mathbf{Y}^{(t)})\mathbf{Q}^{(t)} + \beta\mathbf{A}\mathbf{P}^{(s)} + \lambda\mathbf{L}_P^{-}\mathbf{P}^{(t)}}{\mathbf{I}^{(t)} \odot (\mathbf{P}^{(t)}\mathbf{Q}^{(t)^T})\mathbf{Q}^{(t)} + \alpha\mathbf{P}^{(t)} + \beta\mathbf{A}\mathbf{A}^T\mathbf{P}^{(t)} + \lambda\mathbf{L}_P^{+}\mathbf{P}^{(t)}}},\end{aligned}$$

where $\mathbf{L}_P = \mathbf{L}_P^{+} - \mathbf{L}_P^{-}$, $[\mathbf{L}_P^{+}]_{ij} = (|[\mathbf{L}_P]_{ij}| + [\mathbf{L}_P]_{ij})/2 \geq 0$ and $[\mathbf{L}_P^{-}]_{ij} = (|[\mathbf{L}_P]_{ij}| - [\mathbf{L}_P]_{ij})/2 \geq 0$.

Similarly, let $\mathbf{B}$ be the item mapping matrix whose element $\mathbf{B}_{i,g_i} = 1$ if $(i, g_i) \in \mathcal{A}^{(t,s)}$, zero otherwise. We can obtain the following updating rules from the derivatives of $\mathcal{J}$ w.r.t $\{Q^{(\pi)}\}$:

$$\begin{aligned}
\mathbf{Q}^{(s)} &= \mathbf{Q}^{(s)} \odot \\
&\sqrt{\frac{(\mathbf{I}^{(s)} \odot \mathbf{Y}^{(s)})^T \mathbf{P}^{(s)} + \beta \mathbf{B}^T \mathbf{Q}^{(t)}}{(\mathbf{I}^{(s)} \odot (\mathbf{P}^{(s)} \mathbf{Q}^{(s)^T}))^T \mathbf{P}^{(s)} + \alpha \mathbf{Q}^{(s)} + \beta \mathbf{B}^T \mathbf{B} \mathbf{Q}^{(s)}}}, \\
\mathbf{Q}^{(t)} &= \mathbf{Q}^{(t)} \odot \\
&\sqrt{\frac{(\mathbf{I}^{(t)} \odot \mathbf{Y}^{(t)})^T \mathbf{P}^{(t)} + \beta \mathbf{B}^T \mathbf{Q}^{(s)} + \lambda \mathbf{L}_Q^- \mathbf{Q}^{(t)}}{(\mathbf{I}^{(t)} \odot (\mathbf{P}^{(t)} \mathbf{Q}^{(t)^T}))^T \mathbf{P}^{(t)} + \alpha \mathbf{Q}^{(t)} + \beta \mathbf{B}^T \mathbf{B} \mathbf{Q}^{(t)} + \lambda \mathbf{L}_Q^+ \mathbf{Q}^{(t)}}},
\end{aligned}$$

where $\mathbf{L}_Q = \mathbf{L}_Q^+ - \mathbf{L}_Q^-$, $[\mathbf{L}_Q^+]_{ij} = (|[\mathbf{L}_Q]_{ij}| + [\mathbf{L}_Q]_{ij})/2 \geq 0$ and $[\mathbf{L}_Q^-]_{ij} = (|[\mathbf{L}_Q]_{ij}| - [\mathbf{L}_Q]_{ij})/2 \geq 0$.

Given the learned latent factors, the missing ratings in $\mathbf{Y}^{(t)}$ are predicted as $\hat{\mathbf{Y}}^{(t)} = \mathbf{P}^{(t)} \mathbf{Q}^{(t)^T}$.

3.5 Experiments

3.5.1 Experimental Setup

We evaluate our proposed recommendation model on two real-world datasets: Yelp[1] and Epinions (Tang et al., 2012). For both datasets, we filter out all the locations/items with less than 5 ratings and all the users who only have 1 rating. The statistics of the datasets after filtering are listed in Table VII.

The Epinions dataset is used for testing the scenario of an emerging online business that shares some users with a developed online business. As in (Li and Lin, 2014), we partition the

[1] http://www.yelp.com/dataset_challenge/

TABLE VII

Statistics of Datasets

Name	#users	#items/locations	#tags	#ratings	#social links
Epinions	$21,740$	$31,678$	27	$541,108$	$344,286$
Yelp	$26,618$	$8,467$	770	$230,418$	$183,765$

dataset into two parts with partially overlapping users and items. One part consists of ratings given by 80% users, which serves as the source business. The other part consists of ratings given by 20% users, which serves as the target business. In this task, 20% users and all the items in the target business are overlapping with the source business. The Yelp dataset is used for testing the scenario of a mature online business that starts to branch into new geographic areas. There are $176,736$ ratings on $6,317$ locations given by $19,464$ users in Arizona, and $53,682$ ratings on $2,150$ locations given by $11,476$ users in Nevada. We consider Arizona as the source business and Nevada as the target business. $4,322$ users are shared in both businesses but the locations are disjoint.

For each target business, we conduct the experiments using 5-fold cross-validation and report the average results of 5-fold cross validation on the datasets. Since we are interested in the cold start problem, we select the cold start users, who have less than 5 ratings in the training set, in each fold and report their results separately. We use “Mean Absolute Error” (MAE) and the “Root Mean Square Error” (RMSE) metrics to measure the prediction quality. Both metrics have been widely used in the recommendation problem (Zhang et al., 2006; Yu et al., 2013; Luo et al., 2014) and the smaller is the value of each criterion, the better is the performance.

TABLE VIII

Augmented meta paths used in the experiment.

Intra-network augmented meta path
$(C \xrightarrow{[star]} P \xleftarrow{[star]} C)$,$(P \xleftarrow{[star]} C \xrightarrow{[star]} P)$, $(P \rightarrow T \leftarrow P)$
$(C \xrightarrow{[star]} P \xleftarrow{[star]} C \xrightarrow{[star]} P \xleftarrow{[star]} C)$
$(P \xleftarrow{[star]} C \xrightarrow{[star]} P \xleftarrow{[star]} C \xrightarrow{[star]} P)$
Inter-network augmented meta path
$(P \leftrightarrow I \xleftarrow{[score]} U \xrightarrow{[score]} I \leftrightarrow P)$
$(C \leftrightarrow U \xrightarrow{[score]} I \xleftarrow{[score]} U \leftrightarrow C)$, $(C \leftrightarrow U \leftrightarrow U \leftrightarrow C)$
C and P denote “customer” and “product” in $G^{(t)}$, respectively. U, I and T denote “user”, “item” and “tag” in $G^{(s)}$, respectively. $\leftrightarrow$ denotes the anchor links across $G^{(s)}$ and $G^{(t)}$.

3.5.2 Comparison methods

We compare our proposed model with several state-of-the-art recommendation models. First, we implement the models that utilize heterogeneous relationships to capture similarity information as geometric regularization for improving recommendation performance. Weighted nonnegative matrix factorization (**WNMF** (Zhang et al., 2006)) on the target rating matrix is utilized as the baseline method. In the following methods, different similarity measures are introduced for constructing the geometric regularization and incorporating into the WNMF model.

- **Hete-MF** (Yu et al., 2013): uses *PathSim* to measure the similarity between items.
- **Hete-CF** (Luo et al., 2014): extends Hete-MF by also considering the relationship between users.

- **Hete-PRW**: We implement *pairwise random walk* to measure the similarity between users/items.
- **Amp-MF**: We use our proposed similarity measure *AmpSim* to compute the similarity between users/items.

In the experiments, we construct aligned networks for both datasets based on the network schema in Figure 7 and utilize eight different similarity semantics listed in Table VIII to capture the similarity information for the above models. For simplicity, the weights of different AMPs in Equation 3.2 are assigned with identical values, i.e., $\omega = [\frac{1}{8}, , \frac{1}{8}]$.

Next, we compare with the models that learn common latent factors from multiple relative matrices.

- **CMF** (Singh and Gordon, 2008): Collective matrix factorization with alignment regularization is used as the state-of-the-art transfer learning model with cross-domain entity correspondences.
- **RMGM** (Li et al., 2009b): Rating-matrix generative modelis used as the state-of-the-art model without entity correspondences.
- **Amp-CMF**: Our proposed recommendation model.

Parameters of baselines are set to be consistent with the values recommended in their corresponding papers. For each method, we randomly initialize the latent factors and set the maximum number of iterations as 100. For **RMGM**, the latent dimensionality k is set as the number of clusters, the default choice for most kernel-based approaches. The sparsity tradeoff

TABLE IX

Performance comparison (mean±std) on the Epinion dataset

Method	**Overall**				**Cold Start**			
	RMSE		**MAE**		**RMSE**		**MAE**	
	$k = 10$	$k = 20$	$k = 10$	$k = 20$	$k = 10$	$k = 20$	$k = 10$	$k = 20$
WNMF	1.551 ± 0.009	1.533 ± 0.009	1.156 ± 0.008	1.153 ± 0.009	1.596 ± 0.016	1.594 ± 0.016	1.219 ± 0.016	1.218 ± 0.013
Hete-MF	1.402 ± 0.010	1.398 ± 0.009	1.034 ± 0.009	1.030 ± 0.008	1.462 ± 0.017	1.454 ± 0.014	1.106 ± 0.016	1.101 ± 0.013
Hete-CF	1.148 ± 0.008	1.141 ± 0.008	0.908 ± 0.007	0.901 ± 0.007	1.211 ± 0.011	1.201 ± 0.011	0.961 ± 0.009	0.955 ± 0.009
Hete-PRW	1.395 ± 0.009	1.392 ± 0.009	1.039 ± 0.005	1.030 ± 0.005	1.434 ± 0.009	1.428 ± 0.009	1.072 ± 0.005	1.066 ± 0.005
Amp-MF	**1.099±0.009**	**1.097±0.009**	**0.869±0.005**	**0.868±0.005**	**1.131±0.009**	**1.128±0.009**	**0.899±0.005**	**0.897±0.005**
CMF	1.152 ± 0.007	1.143 ± 0.007	0.870 ± 0.004	0.868 ± 0.005	1.198 ± 0.012	1.185 ± 0.010	0.902 ± 0.009	0.899 ± 0.009
RMGM	1.246 ± 0.008	1.242 ± 0.010	0.989 ± 0.005	0.983 ± 0.007	1.271 ± 0.005	1.266 ± 0.009	1.013 ± 0.002	1.014 ± 0.006
Amp-CMF	**1.097±0.009**	**1.095±0.009**	**0.867±0.005**	**0.866±0.005**	**1.129±0.009**	**1.127±0.009**	**0.898±0.005**	**0.896±0.005**

parameter α is fixed as 0.1 for both datasets. We set the similarity tradeoff parameter $\lambda = 1$ as in (Yu et al., 2013; Luo et al., 2014) and tune the alignment tradeoff parameter by searching the grid of $\{10^{-5}, \cdots, 10^{3}\}$.

3.5.3 Performance Analysis

Table IX and Table X present the RMSE and MAE results of all methods, with different dimensionality settings ($k = 10$ and 20), on both datasets. The column Overall and Cold Start report the performance on the whole test user set and the cold start user set, respectively. The best results in each experimental setting are listed in bold.

We first investigate the performance of the models with geometric regularization. It can be seen that Hete-CF performs better than WNMF and Hete-MF, likely due to the fact that incorporating more similarity information can alleviate the data sparsity issue and improve the prediction accuracy. For all cases, Amp-MF beats all baseline methods. We believe this is because the more accurate similarity measure can usually lead to the better recommendation

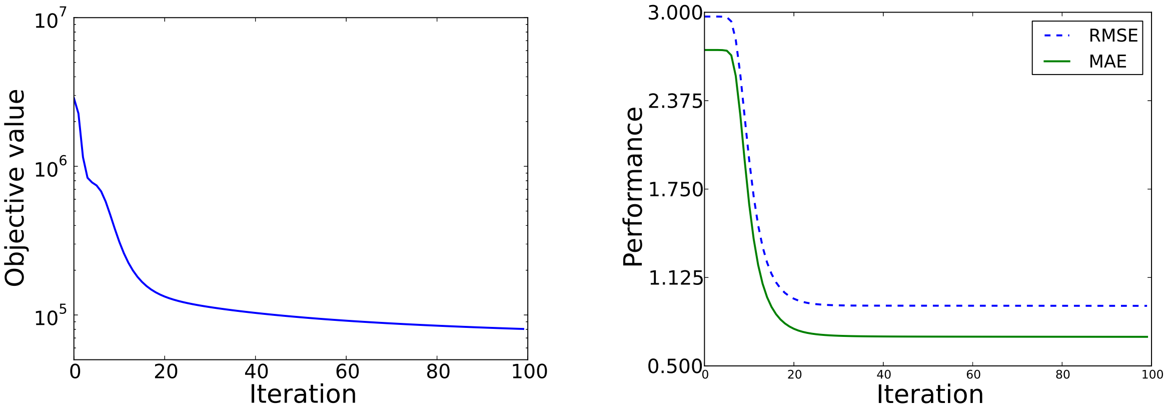
TABLE X

Performance comparison (mean±std) on the Yelp dataset

Method	**Overall**				**Cold Start**			
	RMSE		**MAE**		**RMSE**		**MAE**	
	$k = 10$	$k = 20$	$k = 10$	$k = 20$	$k = 10$	$k = 20$	$k = 10$	$k = 20$
WNMF	1.446 ± 0.009	1.429 ± 0.009	1.097 ± 0.006	1.083 ± 0.005	1.535 ± 0.014	1.520 ± 0.013	1.184 ± 0.005	1.170 ± 0.005
Hete-MF	1.429 ± 0.009	1.351 ± 0.009	1.086 ± 0.005	1.006 ± 0.005	1.518 ± 0.012	1.492 ± 0.011	1.171 ± 0.005	1.148 ± 0.005
Hete-CF	1.305 ± 0.008	1.199 ± 0.008	0.957 ± 0.005	0.907 ± 0.005	1.378 ± 0.009	1.228 ± 0.009	1.017 ± 0.005	0.935 ± 0.005
Hete-PRW	1.343 ± 0.008	1.313 ± 0.008	1.018 ± 0.005	0.991 ± 0.005	1.414 ± 0.008	1.382 ± 0.008	1.088 ± 0.005	1.059 ± 0.005
Amp-MF	**1.191±0.009**	**1.187±0.009**	**0.899±0.005**	**0.897±0.005**	**1.219±0.008**	**1.215±0.008**	**0.928±0.005**	**0.925±0.005**
CMF	1.294 ± 0.009	1.274 ± 0.010	0.966 ± 0.005	0.949 ± 0.006	1.349 ± 0.012	1.329 ± 0.012	1.015 ± 0.005	0.998 ± 0.006
RMGM	1.240 ± 0.009	1.238 ± 0.009	0.925 ± 0.005	0.902 ± 0.004	1.316 ± 0.009	1.295 ± 0.009	0.995 ± 0.004	0.974 ± 0.004
Amp-CMF	**1.134±0.009**	**1.127±0.009**	**0.854±0.005**	**0.847±0.005**	**1.148±0.009**	**1.139±0.009**	**0.875±0.006**	**0.865±0.006**

performance. Our proposed similarity measure, AmpSim, is capable of leveraging the relationship attributes to find the most similar entities.

Next, we compare the models that learn common latent factors from multiple relative matrices. CMF preforms better than RMGM in the Epinions dataset while the results are opposite in the Yelp dataset. The major reason is that CMF requires entity correspondences for knowledge transfer. In the Epinions dataset, the items in the target business are contained in the source business, but in the Yelp dataset the items (locations) are disjoint. The lack of overlapping items makes CMF not able to transfer the common information of items from the source business to the target business. Furthermore, we can see that Amp-CMF consistently outperform all the other comparison models. This confirms our assumption that integrating the similarity information with the CMF model would enjoy the intrinsic mutual reinforcement and boost the recommendation quality.



(a) Objective function

(b) Performance on the training sets

Figure 8. Converge Rate on the Yelp Dataset

In general, most of the models with higher dimensionality (K=20) perform slightly better than that with lower dimensionality (K=10). Besides, all models have higher prediction errors on the cold start user sets, who have less than 5 ratings the training set, than on the whole testing user sets. Noteworthily, Amp-CMP also outperforms all the other baseline methods for the cold start user sets. This confirms the effectiveness of Amp-CMF for cold start recommendation.

Since Amp-CMF is an iterative algorithm based on CMF, we need to check its convergence property empirically. Figure 8 presents the convergence process of Amp-CMF in terms of the objective function (in Equation 3.3) and the training performance. From 8(a), one can claim that the proposed objective function converges nicely and the objective value can be reduced rapidly in the first few iterations. From Figure 8(b), we notice that it takes a small number of iterations before the error rates start to converge. This is because the latent factors of similar entities are enforced to be close, while their initial values are randomly allocated (as in most matrix factorization based algorithms). It needs a few iterations as a initialization phase before the latent factors start to fit the training sets.

3.6 Related Work

Transfer learning based on CMF has been proposed to utilized information shared by related domains to learn latent factors for better recommendations. The first category of such methods assume that there are certain overlapping users and/or items across multiple domains and align the latent factors of the overlapping users and/or items (Singh and Gordon, 2008; Long et al., 2010). Another approach is to enforce the similarity of cluster-level preference patterns in two related domains, without assuming cross-domain entity correspondences. Codebook-based-transfer (CBF) (Li et al., 2009a) proposed to align the two kernel matrices of users from two domains using a co-clustering process. Rating-matrix generative model (RMGM) uses a probabilistic graphical model to relax the constraints in CBT from hard clustering to soft clustering. Although the former approach is more effective for knowledge transfer in general, it can be biased to the overlapping portions. One possible way to handle the above issue is to preserve geometric closeness between similar entities (Wang and Mahadevan, 2011; Long et al., 2014). However, it may not be trivial to find reliable similarity information in the emerging domain.

When building a recommendation system, it is crucial to choose a good way to measure the entity similarity. A number of studies leverage linkage structure of a network for measuring similarity, e.g., *personalized PageRank* (Jeh and Widom, 2003) and *SimRank* (Jeh and Widom, 2002), while they are defined on a homogeneous network or a bipartite network. However, in real-world applications, there are lots of heterogeneous networks. Meta path was introduced in (Sun et al., 2011; Shi et al., 2012) to measure the similarity in heterogeneous networks. A series of meta path-based similarity measures are proposed for either the same type of entities (Sun et al., 2011) or different type of entities (Shi et al., 2014; Cao et al., 2014b; Shi et al., 2015). Recently, researchers have been aware of the importance of heterogeneous information for recommendation. (Yu et al., 2013) proposed an implicit feedback recommendation model with the similarity information extracted from heterogeneous network. (Luo et al., 2014) proposed a CF-based social recommendation method, called Hete-CF, using heterogeneous relations. (Vahedian, 2014) proposed the WHyLDR approach for multiple recommendation tasks. It combines a linear-weighted hybrid model with heterogeneous information. These methods usually treat the relationships between entities as binary connections. However, plentiful attributes are usually attached to the relationships, e.g., the rating of an item given by a user and the timestamp of a post. As demonstrated in the experiments, discarding these important attributes can lead to a degraded performance.

CHAPTER 4

MULTI-TASK MULTI-VIEW LEARNING

(This chapter was previously published as "Multilinear Factorization Machines for Multi-Task Multi-View Learning", in WSDM '17 (Lu et al., 2017). DOI: `http://dx.doi.org/10.1145/3018661.3018716`.)

4.1 Introduction

In the era of big data, it is becoming common to have heterogeneous data obtained in multiple views or extracted from multiple sources, known as "multi-view data". For example, in web image retrieval, the visual information of images and their textual descriptions can be regarded as two views; for scientific document categorization, each paper has word features and its citations. Multi-View Learning (MVL) was proposed to combine different views to obtain better performance than relying on just one single view (Cao et al., 2016; Sindhwani and Rosenberg, 2008; Xu et al., 2013; Varma and Babu, 2009). In addition, different learning tasks might be related with each other through shared features. For example, for product recommendation systems, different product domains can be viewed as different tasks and they might share certain word features in product reviews, *e.g.*, good, great and bad. Multi-Task Learning (MTL) was developed to learn multiple related tasks together instead of learning them separately, so as to improve the performance of each task (Ando and Zhang, 2005; Argyriou et al., 2008; Chen et al., 2009; Chen et al., 2011; Evgeniou and Pontil, 2007; Evgeniou and Pontil, 2004; Gong et al., 2012).

Existing MVL or MTL approaches only capture one type of heterogeneity, while real-world problems exhibit dual-heterogeneity (both feature heterogeneity and task heterogeneity) (He and Lawrence, 2011; Jin et al., 2013; Zhang and Huan, 2012). Consider the multi-task recommendation problem, reviews may also contain multi-view data such as words, emotion icons, images and web links to other products. Such type of problem is Multi-Task Multi-View (MTMV) learning. Though there are a wide variety of applications of MTMV learning, only a few works have addressed this problem. Recently, a graph-based iterative algorithm, IteM^2 (He and Lawrence, 2011), was first introduced for MTMV learning with applications to text classification, while it can only deal with nonnegative feature values. Assuming the predictive models should be consistent among different views, co-regularization based algorithms, such as regMVMT (Zhang and Huan, 2012) and CSL-MTMV (Jin et al., 2013), imposed a regularization term to enforce the difference of the predictive functions among different views to be small.

Nonetheless, different views might not be consistent with each other especially for heterogeneous data; instead, they provide complementary information. For example, textual view (*e.g.*, bag-of-words) and topological view (*e.g.*, citations in the document categorization or web links in the product recommendation) usually provide complementary information. In contrast to building a distinct model for each view or each task, another direction is to mine the hidden interactions/correlations among MTMV features.

In this chapter, we propose a general framework for learning the predictive multilinear structure from the complex relationships within the heterogeneous data. Specifically, we model the multimodal interactions among multiple tasks and multiple views as a tensor structure, by taking the tensor product of their respective feature spaces. As the interactions with different orders can reflect different but complementary insights (Cao et al., 2016; Rendle, 2010), in the proposed framework, the full-order interactions observed in the heterogeneous data are used collectively to learn the consensus representation. In this manner, it can also deal with the partially incomplete data without difficulty because the learning procedure does not simply rely on any particular view.

Constructing the full-order tensor may not be realistic for real-world applications, as the model parameters can be exponential growth. We further propose multilinear factorization machines (MFMs) that can efficiently learn the task-specific feature map and the task-view shared multilinear structures, without physically building the tensor. A joint factorization is applied which makes parameter estimation more accurate under sparsity and avoid overfitting. Furthermore, the model complexity of the proposed MFMs is linear in the feature dimensionality, making it applicable to large-scale real-world applications.

The contributions of this work are summarized as follows:

- The proposed framework is widely applicable to multiple types of heterogeneous machine learning problems, by learning predictive multilinear structures from the complex relationships within the heterogeneous data.

TABLE XI

List of basic symbols.

Symbol	Definition and description
x	a scale is denoted by a lowercase letter
$\mathbf{x}$	a vector is denoted by a boldface lowercase letter
$\mathbf{X}$	a matrix is denoted by a boldface uppercase letter
$\mathcal{X}$	a tensor, set or space is represented by a calligraphic letter
$[1 : M]$	denotes a set of integers in the range of 1 to M inclusively.
$\langle \cdot, \cdot \rangle$	denotes inner product
$\circ$	denotes tensor product (outer product)
$*$	denotes Hadamard (element-wise) product

- We propose multilinear factorization machines (MFMs) that can efficiently learn the task-specific feature map and the task-view shared multilinear structures from full-order interactions.
- Extensive experiments on four real-world datasets demonstrate that the proposed MFM methods outperform several state-of-the-art methods in a wide variety of MTMV problems, including classification tasks and regression tasks.

4.2 Preliminaries

The key to this work is to apply the tensor structure to fuse all possible dependence relationships among different views and different tasks. We begin by introducing some related concepts and notation about tensor, and then state the problem of multi-task multi-view learning. Table XI lists basic symbols that will be used throughout the chapter.

4.2.1 Tensor Basics and Notation

Tensors is a mathematical representation of higher order arrays. Following (Kolda and Bader, 2009), an M-th order tensor is denoted by $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ and its elements by $x_{i_1, \cdots, i_M}$. An index is denoted by a lowercase letter, spanning in the range of $[1, 2, \cdots, I]$. All vectors are column vectors unless otherwise specified. For an arbitrary matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$, the i-th row is denoted by $\mathbf{x}^i$ and the j-th column vector is denoted by $\mathbf{x}_j$. The inner product of two same-sized tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ is defined by $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \cdots \sum_{i_1=1}^{I_M} x_{i_1, \cdots, i_M} y_{i_1, \cdots, i_M}$. The outer product of M vectors $\mathbf{x}^{(m)} \in \mathbb{R}^{I_m}$ for $m \in [1 : M]$ is an M-th order tensor and defined elementwise by $\left(\mathbf{x}^{(1)} \circ \cdots \circ \mathbf{x}^{(M)}\right)_{i_1, \cdots, i_M} = x_{i_1}^{(1)} \cdots x_{i_M}^{(M)}$ for all values of the indices. In particular, given $\mathcal{X} = \mathbf{x}^{(1)} \circ \cdots \circ \mathbf{x}^{(M)}$ and $\mathcal{Y} = \mathbf{y}^{(1)} \circ \cdots \circ \mathbf{y}^{(M)}$, it holds that

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \prod_{m=1}^{M} \langle \mathbf{x}^{(m)}, \mathbf{y}^{(m)} \rangle = \prod_{m=1}^{M} \mathbf{x}^{(m)\mathrm{T}} \mathbf{y}^{(m)} \tag{4.1}$$

For a general tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$, its CP factorization is

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{x}_r^{(1)} \circ \cdots \circ \mathbf{x}_r^{(M)} = [\![\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(M)}]\!] \tag{4.2}$$

where for $m \in [1 : M]$, $\mathbf{X}^{(m)} = [\mathbf{x}_1^{(m)}, \cdots, \mathbf{x}_R^{(m)}]$ are factor matrices of size $I_m \times R$, R is the number of factors, and $[\![\cdot]\!]$ is used for shorthand.

4.2.2 Problem Formulation

Suppose that the problem includes T tasks and V views. Let N_t be the number of labeled instances in the task $t \in [1 : T]$, thus we have $N = \sum_t N_t$ labeled instances in total. Let I_v

be the dimensionality of the view $v \in [1 : V]$ and denote $I = \sum_v I_v$. Assuming the problem is associated with training data $\mathcal{D} = \big\{(\mathbf{X}_t^{(1)}, \ldots, \mathbf{X}_t^{(V)},\ \mathbf{y}_t) \mid t \in [1 : T]\big\}$, where $\mathbf{X}_t^{(v)} \in \mathbb{R}^{I_v \times N_t}$ is the feature matrix in the t-th task for the v-th view; $\mathbf{y}_t$ is the vector of the responses of the t-th task. The goal is to leverage the discriminative information from all tasks, as well as the complementary among different views, to make prediction on the unlabeled instances in each task. Specifically, for the t-th task, we are interested in finding a predictive function $f_t : \mathcal{X}_t \to \mathcal{Y}_t$ that minimizes the expected loss, where $\mathcal{X}_t$ is the input space and $\mathcal{Y}_t$ is the output space.

The objective is to learn T functions $\{f_t\}_{t=1}^T$ that minimize the following regularized empirical risk:

$$\mathcal{R}(\{f_t\}_{t=1}^T) = \sum_{t=1}^{T} \left(\sum_{n=1}^{N_t} \frac{1}{N_t} \ell\left(f_t(\{\mathbf{x}_{t,n}^{(v)}\}), y_{t,n}\right) + \lambda \Omega(f_t) \right) \tag{4.3}$$

where ℓ is a prescribed loss function, Ω is the regularizer encoding the prior knowledge of f_t, and $\lambda > 0$ is the regularization hyperparameter that controls the trade-off between the empirical loss and the prior knowledge. In this formulation, we weight each task equally (by dividing the number of instances N_t) so that no task will dominate the others. One may also choose other weighting schemes. The empirical loss of the training data in the t-th task is

$$\mathcal{L}_t(f_t(\{\mathbf{X}_t^{(v)}\}), \mathbf{y}) = \frac{1}{N_t} \sum_{n=1}^{N_t} \ell\left(f_t(\{\mathbf{x}_{t,n}^{(v)}\}), y_{t,n}\right) \tag{4.4}$$

The choice of the loss function ℓ depends on learning tasks. To conduct regression, for example, one can use the squared loss, and for classification problems, one can use the logistic loss or cross-entropy. The formulation of regularizer is chosen based on prior knowledge about data.

For solving MTMV problems, a straightforward approach is to concatenate feature vectors from different views and apply the multi-task learning algorithms. However, this approach would fail to leverage the underlying correlations between different views, wherein complementary information is contained. Through the employment of a nonlinear kernel, such as "polynomial kernel" or "RBF kernel", one can implicitly project data samples from the feature space into a high-dimensional space. This allows modeling the higher order interactions between features. However, the interaction parameters are learned completely independent (Rendle, 2010). Moreover, the inter-view correlations are evaluated only at the view-level, while the explicit correlations between features among different views are failed to be explored.

Currently, there are only a few researches on MTMV learning. The IteM^2 algorithm (He and Lawrence, 2011) projects any two tasks to a new RKHS based on the common views shared by the given two tasks, and has been shown empirically outperforming multiple kernel approaches. Assuming the predictive models should be consistent among different views, co-regularization based methods were later developed (Jin et al., 2013; Zhang and Huan, 2012). These methods assume that all the views are similar to each other, while such assumption may not be appropriate especially for heterogeneous data. Furthermore, the co-regularization approaches involve pairwise comparison of the prediction from different views, which leads to high model complexity (the space and time complexity are quadratic and cubic in the total number of features, respectively) (Jin et al., 2013; Zhang and Huan, 2012). Thus, they can hardly be applied to high-dimensional data.

In the following, we will introduce a general framework that intrinsically models the complex relationships in multimodal interactions among multiple tasks and multiple views as a tensor structure.

4.3 Multilinear Structure Learning for MTMV Problems

In this section, we first discuss how to design the multilinear predictive models for learning the full-order interactions among MTMV data. We then derive efficient Multilinear Factorization Machines (MFMs) that can learn the shared multilinear structure in linear complexity.

4.3.1 Multilinear Predictive Models

Given an input vector $\mathbf{x} \in \mathbb{R}^I$, the linear model for the t-th task is given by

$$f_t(\mathbf{x}) = \sum_{i=1}^{I} w_t x_i = \mathbf{x}^{\mathrm{T}} \mathbf{w}_t \tag{4.5}$$

where $\mathbf{w}_t \in \mathbb{R}^I$ is the weight vector for linear effects. Let $\mathbf{W} \in \mathbb{R}^{I \times T}$ denote the weight matrix to be learned, whose columns are the vector $\mathbf{w}_t$. Most of the MTL algorithms aim at solving the following regularized problem

$$\min \sum_{t=1}^{T} \mathcal{L}_t \left(\mathbf{X}_t^{\mathrm{T}} \mathbf{w}_t, \mathbf{y}_t \right) + \lambda \Omega(\mathbf{W}) \tag{4.6}$$

Many different assumptions about how tasks are related have been proposed for MTL, leading to different regularization terms (e.g., ℓ_1/ℓ_q-norm regularization, trace norm regularization, and composite regularization) in the formulation (Ando and Zhang, 2005; Argyriou et al., 2008; Chen et al., 2009; Chen et al., 2011; Gong et al., 2012).

In fact, the joint learning of multiple linear models for multiple tasks is essentially to learn the bilinear map for modeling the second-order interactions between input features and tasks. Let $\mathbf{e}_t \in \mathbb{R}^T$ denote the task indicator vector

$$\mathbf{e}_t = [\underbrace{0, \cdots, 0}_{\text{t-1}}, 1, 0, \cdots, 0]^{\mathrm{T}}$$

We then have that

$$f_t(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{w}_t = \mathbf{x}^{\mathrm{T}}\mathbf{W}\mathbf{e}_t = \langle \mathbf{W}, \mathbf{x} \circ \mathbf{e}_t \rangle = f(\{\mathbf{x}, \mathbf{e}_t\}) \tag{4.7}$$

Similarly, we can form a multilinear function for modeling the higher-order interactions in MTMV data. Assume we are given two views, for example, we can learn the third-order interactions by

$$f_t(\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\}) = \mathbf{x}^{(1)^{\mathrm{T}}}\mathbf{W}_t\mathbf{x}^{(2)} = \left\langle \mathcal{W}, \mathbf{x}^{(1)} \circ \mathbf{x}^{(2)} \circ \mathbf{e}_t \right\rangle \tag{4.8}$$

where $\mathcal{W} \in \mathbb{R}^{I_1 \times I_2 \times T}$ is the weight tensor to be learned.

However, only the highest-order interactions are explored in this way, and such interactions are limited in sparse data, especially when one or more views are missing in some instances. In contrast, the lower-order (*e.g.*, pairwise) interactions can usually explain the data sufficiently (Rendle and Schmidt-Thieme, 2010; Rendle, 2010), and incorporating the lower-order interac-

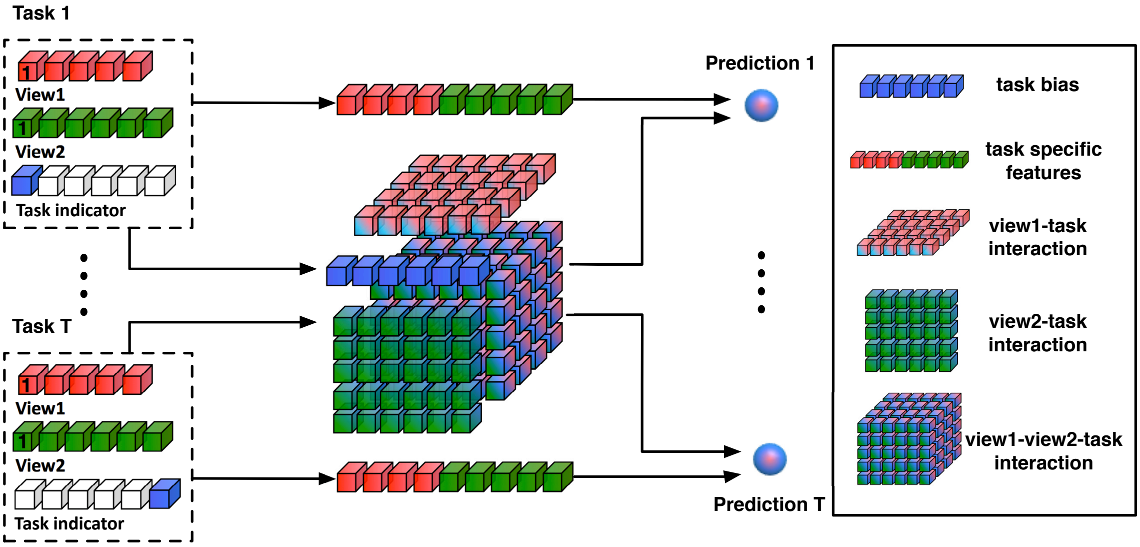


Figure 9. Multilinear Predictive Models. The feature interactions in multi-task multi-view data are modeled in a full-order tensor. The prediction is learned from the task-specific feature space and the task-view shared multilinear feature space.

tions in the predictive models can further improve the performance (Cao et al., 2016; Rendle, 2010). Hence, we consider nesting all interactions up to full-order:

$$f_t(\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\}) = w_t + \sum_{v=1}^{2} \mathbf{x}^{(v)^{\mathrm{T}}} \mathbf{w}_t^{(v)} + \mathbf{x}^{(1)^{\mathrm{T}}} \mathbf{W}_t \mathbf{x}^{(2)} \tag{4.9}$$

This can be done by adding a dummy value 1 to the feature vector $\mathbf{x}_t^{(v)}$, *i.e.*, $\mathbf{z}_t^{(v)} = [1; \mathbf{x}_t^{(v)}] \in \mathbb{R}^{1+I_v}$. Then Equation 4.9 can be rewritten as

$$f_t(\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\}) = \left\langle \mathcal{W}, \mathbf{z}^{(1)} \circ \mathbf{z}^{(2)} \circ \mathbf{e}_t \right\rangle = \langle \mathcal{W}, \mathcal{Z}_t \rangle \tag{4.10}$$

We can easily extend Equation 4.10 to the MTMV problems with more views. Formally, let $\mathcal{Z}_t = \mathbf{z}^{(1)} \circ \cdots \circ \mathbf{z}^{(V)} \circ \mathbf{e}_t \in \mathbb{R}^{(1+I_1)\times\cdots\times(1+I_V)\times T}$ be the full-order tensor, and let $\mathcal{W} = \{w_{i_1,\ldots,i_V,t}\} \in \mathbb{R}^{(1+I_1)\times\cdots\times(1+I_V)\times T}$ be the weight tensor to be learned. The multilinear map function can be defined as

$$f_t(\{\mathbf{x}^{(v)}\}) = \langle \mathcal{W}, \mathcal{Z}_t \rangle = \sum_{s=1}^{T} \sum_{i_1=0}^{I_1} \cdots \sum_{i_V=0}^{I_V} w_{i_1,\ldots,i_V,s} \left(e_{t,s} \prod_{v=1}^{V} z_{i_v}^{(v)} \right) \tag{4.11}$$

It is worth noting that $w_{i_1,\ldots,i_V,s}$ with some indexes satisfying $i_v = 0$ encodes lower-order interactions between views whose $i_{v'} > 0$.

So far, multiple tasks with multi-view features are able to be incorporated into an elegant tensor formulation, where the complex multiple relationships among tasks and views are embedded within the tensor structures. However, it could be too restrictive to constrain all tasks to share a common set of features (Chen et al., 2011; Gong et al., 2012). Thus, following the structural learning framework for MTL (Ando and Zhang, 2005; Chen et al., 2009), we consider learning a predictive function from both the original feature spaces and the multilinear feature interaction space:

$$f_t(\{\mathbf{x}^{(v)}\}) = \mathbf{x}^{\mathrm{T}}\mathbf{u}_t + \langle \mathcal{W}, \mathcal{Z}_t \rangle \tag{4.12}$$

where $\mathbf{x} = [\mathbf{x}^{(1)}; \ldots; \mathbf{x}^{(V)}] \in \mathbb{R}^I$ is the concatenated feature vector from multiple views, and $\mathbf{u}_t \in \mathbb{R}^I$ is the task-specific weight vector. Figure 9 illustrates the proposed multilinear predictive model with two views.

4.3.2 Multilinear Factorization Machines

Directly learning the weight tensor $\mathcal{W}$ leads to two drawbacks. First, transfer learning is not possible straight from the model since the weight parameters are learned independently for different tasks and different views. Second, the number of parameters in Equation 4.11 is $T\prod_{v=1}^{V}(1+I_v)$, which is exponential to the number of features and make it easy to overfitting and ineffective on sparse data. Hence, we assume the interaction parameters has a low rank and $\mathcal{W}$ can be factorized as

$$\mathcal{W} = [\![\mathbf{\Theta}^{(1)}, \ldots, \mathbf{\Theta}^{(V)}, \mathbf{\Phi}]\!]$$

by CP factorization. The factor matrix $\mathbf{\Theta}^{(v)} \in \mathbb{R}^{(1+I_v)\times R}$ is the shared structure matrix for the v-th view and the t-th row $\boldsymbol{\phi}^t$ within $\mathbf{\Phi}$ is the task specific weight vector for the t-th task. Then Equation 4.11 is transformed into

$$\begin{aligned}
\langle \mathcal{W}, \mathcal{Z}_t \rangle &= \sum_{s=1}^{T}\sum_{i_1=0}^{I_1}\cdots\sum_{i_V=0}^{I_V}\left(\sum_{r=1}^{R}\phi_{s,r}\prod_{v=1}^{V}\theta_{i_v,r}^{(v)}\right)\left(e_{t,s}\prod_{v=1}^{V}z_{i_v}^{(v)}\right)\\
&= \sum_{r=1}^{R}\left(\sum_{s=1}^{T}\phi_{s,r}e_{t,s}\right)\sum_{i_1=0}^{I_1}\cdots\sum_{i_V=0}^{I_V}\left(\prod_{v=1}^{V}\theta_{i_v,r}^{(v)}z_{i_v}^{(v)}\right) \qquad (4.13)\\
&= \sum_{r=1}^{R}\left\langle \boldsymbol{\theta}_r^{(1)}\circ\cdots\circ\boldsymbol{\theta}_r^{(V)}\circ\boldsymbol{\phi}_r\ ,\ \mathbf{z}^{(1)}\circ\cdots\mathbf{z}^{(V)}\circ\mathbf{e}_t\right\rangle
\end{aligned}$$

Because $e_{t,s} = 1$ only when $t = s$ and according to Equation 4.1, we can further rewrite Equation 4.13 into

$$\begin{aligned} \langle \mathcal{W}, \mathcal{Z}_t \rangle &= \sum_{r=1}^{R} \left(\mathbf{z}^{(1)^{\mathrm{T}}} \boldsymbol{\theta}_r^{(1)} \right) \cdots \left(\mathbf{z}^{(V)^{\mathrm{T}}} \boldsymbol{\theta}_r^{(V)} \right) \phi_{t,r} \\ &= \boldsymbol{\phi}^t \left(\left(\mathbf{z}^{(1)^{\mathrm{T}}} \boldsymbol{\Theta}^{(1)} \right) * \cdots * \left(\mathbf{z}^{(V)^{\mathrm{T}}} \boldsymbol{\Theta}^{(V)} \right) \right)^{\mathrm{T}} \\ &= \boldsymbol{\phi}^t \prod_{v=1}^{V} * \left(\mathbf{z}^{(v)^{\mathrm{T}}} \boldsymbol{\Theta}^{(v)} \right)^{\mathrm{T}} \end{aligned} \tag{4.14}$$

where $*$ is the Hadamard (elementwise) product. It should be noted that the first row $\boldsymbol{\theta}^{(v),0}$ within $\boldsymbol{\Theta}^{(v)}$ is always associated with $z_0^{(v)} = 1$ and represents the bias factors of the v-th view. Through the bias factors, the lower-order interactions are explored in the predictive function.

By replacing the tensor inner product in Equation 4.12 using Equation 4.14, we then have

$$f_t(\{\mathbf{x}^{(v)}\}) = \mathbf{x}^{\mathrm{T}} \mathbf{u}_t + \boldsymbol{\phi}^t \prod_{v=1}^{V} * \left(\mathbf{z}^{(v)^{\mathrm{T}}} \boldsymbol{\Theta}^{(v)} \right)^{\mathrm{T}} \tag{4.15}$$

We name this model as multilinear factorization machines (MFMs). Clearly, the parameters of the interactions between multiple tasks with multiple views are jointly factorized. The joint factorization benefits parameter estimation under sparsity, since dependencies exist when the interactions share the same features. Therefore, the model parameters can be effectively learned without direct observations of such interactions especially in highly sparse data. Further, since the lower-order interactions are modeled with the bias factors, this joint factorization model can easily deal with missing views and even incomplete views for multiple tasks.

Another appealing property of MFMs comes from the main characteristics of multilinear analysis. After factorizing the weight tensor $\mathcal{W}$, there is no need to construct the input tensor physically. From Equation 4.15, we can find that the common subspace shared by multiple views can be discovered through the Hadamard product of the low-dimensional projection of each view, and its contribution to the t-th task is controlled by the weight vector $\boldsymbol{\phi}^t$. Moreover, the model complexity is linear in the number of original features. In particular, the model complexity is $O(R(V + I + T) + \sum_{t=1}^{T} I_t)$, where I_t is the number of features in the t-th task. This multilinear property can help save memory and also speed up the learning procedure.

4.3.3 Learning Multilinear Factorization Machines

Following the regularization formulation in Equation 4.3, we propose to estimate the model parameters by minimizing the following regularized empirical risk:

$$\min \mathcal{R}(\mathbf{\Phi}, \{\mathbf{\Theta}^{(v)}\}, \mathbf{U}) = \sum_{t=1}^{T} \mathcal{L}_t(f_t(\{\mathbf{X}_t^{(v)}\}), \mathbf{y}_t) + \lambda\Omega_\lambda(\mathbf{\Phi}, \{\mathbf{\Theta}^{(v)}\}) + \gamma\Omega_\gamma(\mathbf{U}) \tag{4.16}$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_T] \in \mathbb{R}^{I \times T}$. The regularization Ω_λ and Ω_γ can be Forbenius norm, $\ell_{2,1}$ norm, or other structural regularization. To optimize the objective function, we present the alternating block coordinate descent approach in the rest of this section.

With all other parameters fixed, the minimization over $\mathbf{\Theta}^{(v)}$ simply consists of learning the parameters $\mathbf{\Theta}^{(v)}$ by a regularization method, and the partial derivative of $\mathcal{R}$ w.r.t. $\mathbf{\Theta}^{(v)}$ is given by

$$\frac{\partial \mathcal{R}}{\partial \mathbf{\Theta}^{(v)}} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}_t}{\partial f_t} \frac{\partial f_t}{\partial \mathbf{\Theta}^{(v)}} + \lambda \frac{\partial \Omega_\lambda(\mathbf{\Theta}^{(v)})}{\partial \mathbf{\Theta}^{(v)}} \tag{4.17}$$

where $\frac{\partial \mathcal{L}_t}{\partial f_t} = \frac{1}{N_t} \left[\begin{array}{c} \frac{\partial \ell_{t,1}}{\partial f_t}, \cdots, \frac{\partial \ell_{t,N_t}}{\partial f_t} \end{array} \right]^{\mathrm{T}} \in \mathbb{R}^{N_t}$.

For convenience, we let $\boldsymbol{\pi} \in \mathbb{R}^R$ denote $\prod_{v=1}^{V} * \left(\mathbf{z}^{(v)\mathrm{T}} \mathbf{\Theta}^{(v)}\right)^{\mathrm{T}}$ and $\boldsymbol{\pi}^{(-v)} \in \mathbb{R}^R$ denote $\prod_{v'=1, v' \neq v}^{V} * \left(\mathbf{z}^{(v')\mathrm{T}} \mathbf{\Theta}^{(v')}\right)^{\mathrm{T}}$. Let $\mathbf{\Pi} = [\boldsymbol{\pi}_1, \cdots, \boldsymbol{\pi}_N]^{\mathrm{T}}$ and $\mathbf{\Pi}^{(-v)} = [\boldsymbol{\pi}_1^{(-v)}, \cdots, \boldsymbol{\pi}_N^{(-v)}]^{\mathrm{T}}$. We then have that

$$\begin{aligned} \frac{\partial \mathcal{L}_t}{\partial f_t} \frac{\partial f_t}{\partial \mathbf{\Theta}^{(v)}} &= \frac{1}{N_t} \sum_{n=1}^{N_t} \frac{\partial \ell_{t,n}}{\partial f_{t,n}} \frac{\partial f_{t,n}}{\partial \mathbf{\Theta}^{(v)}} \\ &= \sum_{n=1}^{N_t} \mathbf{z}_{t,n}^{(v)} \left(\left(\frac{1}{N_t} \frac{\partial \ell_{t,n}}{\partial f_{t,n}} \boldsymbol{\phi}^t \right) * \boldsymbol{\pi}_{t,n}^{(-v)} \right) \\ &= \mathbf{Z}_t^{(v)} \left(\left(\frac{\partial \mathcal{L}_t}{\partial f_t} \boldsymbol{\phi}^t \right) * \mathbf{\Pi}_t^{(-v)} \right) \end{aligned} \tag{4.18}$$

With all other variables fixed, the minimization over $\mathbf{\Phi}$ w.r.t the empirical loss simply consists of learning the parameters $\boldsymbol{\phi}^t$ independently. The partial derivative of $\mathcal{R}$ w.r.t. $\mathbf{\Phi}$ is given by

$$\frac{\partial \mathcal{R}}{\partial \mathbf{\Phi}} = \left[\begin{array}{ccc} \frac{\partial \mathcal{L}_1}{\partial f_1} \frac{\partial f_1}{\partial \boldsymbol{\phi}^1} ; & \cdots ; & \frac{\partial \mathcal{L}_T}{\partial f_T} \frac{\partial f_T}{\partial \boldsymbol{\phi}^T} \end{array} \right] + \lambda \frac{\partial \Omega_\lambda(\mathbf{\Phi})}{\partial \mathbf{\Phi}} \tag{4.19}$$

```
Algorithm 2 Learning Multilinear Factorization Machines
Input: Training data $\mathcal{D}$, number of factors $R$, regularization parameter $\lambda, \gamma$ and standard
    deviation $\sigma$
Output: Model parameters $\{\mathbf{\Theta}^{(v)}\}, \mathbf{\Phi}, \mathbf{U}$
  Initialize $\{\mathbf{\Theta}^{(v)}\}, \mathbf{\Phi}, \mathbf{U} \sim \mathcal{N}(0, \sigma)$
  repeat
    for $v := 1$ to $V$ do
      Fixing $\{\mathbf{\Theta}^{(v')}\}_{v' \neq v}$, $\mathbf{\Phi}$ and $\mathbf{U}$, update $\mathbf{\Theta}^{(v)}$
    end for
    Fixing $\{\mathbf{\Theta}^{(v)}\}$ and $\mathbf{U}$, update $\mathbf{\Phi}$
    Fixing $\{\mathbf{\Theta}^{(v)}\}$ and $\mathbf{\Phi}$, update $\mathbf{U}$
  until convergence
```

Following the derivation in Equation 4.18, we have that

$$\frac{\partial \mathcal{L}_t}{\partial f_t}\frac{\partial f_t}{\partial \boldsymbol{\phi}^t} = \left(\frac{\partial \mathcal{L}_t}{\partial f_t}\right)^{\mathrm{T}} \mathbf{\Pi}_t \tag{4.20}$$

With all other variables fixed, the partial derivative of $\mathcal{R}$ w.r.t. $\mathbf{U}$ is given by

$$\frac{\partial \mathcal{R}}{\partial \mathbf{U}} = \left[\begin{array}{ccc} \mathbf{X}_1 \frac{\partial \mathcal{L}_t}{\partial f_t} \ , & \cdots , & \mathbf{X}_T \frac{\partial \mathcal{L}_T}{\partial f_T} \end{array} \right] + \gamma \frac{\partial \Omega_\gamma(\mathbf{U})}{\partial \mathbf{U}} \tag{4.21}$$

where $\mathbf{X}_t = [\mathbf{X}_t^{(1)}; \ldots ; \mathbf{X}_t^{(V)}] \in \mathbb{R}^{I \times N_t}$ is the concatenated feature matrix for the t-th task.

The optimization procedure is summarized in Algorithm 2. In the experiment, we apply AdaGrad (Duchi et al., 2011), an adaptive gradient-based optimization approach that automatically determines a per-parameter learning rate, for parameter updates. The speed bottleneck of this algorithm is in computing the predicted values of all the training instances. The time com-

TABLE XII

The statistics for each dataset.

Classification	#Feature	T	N_p	N_n	Partial View Missing?
FOX	image(996), text(2,711)	4	178∼635	888∼1,345	No
DBLP	linkage(4,638), text(687)	6	635∼1,950	2,688∼3,985	No
Regression	#Feature	T	N	Density	Partial View Missing?
MovieLens	users(943), movies(1,599), tags(1,065)	10	758∼39,895	6.3%	No
Amazon	users(1,805,364), items(192,978), text(83,143)	5	349,038∼1,015,189	0.001%	Yes

plexity of computing Equation 4.15 for an instance is $O(RV+(R+1)I_t))$, while the computation can be done in parallel. It can be further reduced under sparsity, since we only need to compute the sums over the non-zero elements. Let $N_z(\mathbf{x}_{t,n})$ denote the number of non-zero elements, then the time complexity of the computation for each instance is $O(RV+(R+1)N_z(\mathbf{x}_{t,n}))$.

4.4 Experiments

4.4.1 Datasets

To evaluate the performance of the proposed MFMs, we conduct extensive experiments on the following four datasets (two of them are for classification tasks, and two of them are for regression tasks):

- **FOX**[1]: This dataset was crawled from FOX web news (Qian and Zhai, 2014). The category for each news article can be viewed as the class label. For each task (category), the documents from one category are regarded as positive samples. Each instance can be represented in two views, the text view and image view. The text view consists of ℓ_2-

[1] https://sites.google.com/site/qianmingjie/home/datasets/

normalized TF-IDF vector representation extracted from titles, abstracts, and text body contents[1]. The image view consists of seven groups of color features and five textural features.

- **DBLP**[2]: This dataset was extracted from the DBLP (Kong et al., 2011). The research areas (DB, DM, AI, IR, CV, and ML) that an author has published papers in can be viewed as class labels. For each task (research area), the authors who have published papers in that research area are regarded as positive samples. Each instance is represented in two views, the text view and the linkage view. The ℓ_2-normalized TF-IDF vector representation of all the paper titles published by the author is used as the text view data. Since each instance represents an author, the linkage feature of an instance is the binary vector of the co-authors' ids.

- **MovieLens**[3]: Regression tasks for rating prediction are studied on this MovieLens dataset. We select the top 10 genres with the most movies as tasks. Each rating in this dataset has three views, *i.e.*, users, movies and tags. The user view and item view are represented by one-hot encoding. The tags of the movies are used for the tag view.

[1] Stemming, lemmatization, removing stop-words and words with frequency less than 1%, etc., are handled beforehand for all the text view mentioned in this chapter

[2] http://dblp.uni-trier.de/db/

[3] http://grouplens.org/datasets/movielens/

- **Amazon**[1]: This dataset contains item reviews from Amazon (McAuley et al., 2015). We select ratings from the Clothing, Jewelry, Novelty, Accessories and Shoes item categories for the study of large-scale regression tasks, where each category is viewed as a task. Users and items with less than 5 ratings are filtered. Each rating in this dataset has three views, *i.e.*, users, items and text. The user view and item view are constructed using the same way as in the MovieLens dataset. The ℓ_2-normalized TF-IDF vector representation of all summaries of the item is used as the text view.

The statistics for each dataset is summarized in Table XII, where T denotes the number of tasks, N_p, N_n and N denote the number of positive, negative and all the samples in each task, respectively. The density in Table XII means the density of the user-item matrix in the dataset. Note that the text view is partially missing in the Amazon dataset, where 0.62% reviews have no text.

4.4.2 Comparisons

We compare the proposed MFM method with five state-of-the-art methods.

- **Factorization Machine (FM)** explores pairwise interactions between all features. We apply the FM in the setting of MTMV learning by concatenating the task indicator and all the feature vectors from multiple views as the input feature vector. The preliminary study shows it performs better than training each task separately.

[1] http://jmcauley.ucsd.edu/data/amazon/

- **Robust Multi-Task Feature Learning (rMTFL)** is a representative MTL algorithm that uses composite regularization for joint feature learning (Gong et al., 2012).
- **Tensor Factorization (TF)** is a generalization of matrix factorization to higher orders. Since TF only considers the highest-order of the given tensor, we use Equation 4.8 to model the MTMV data for TF and factorize the weight tensor.
- **IteM^2** is a transductive MTMV learning algorithm with applications to classification problems (He and Lawrence, 2011). Since IteM^2 can only handle nonnegative feature values, a positive constant value is added to the feature values such that its nonnegativity can be guaranteed.
- **CSL-MTMV** is an inductive MTMV learning algorithm (Jin et al., 2013) that assumes the predictions of different views within a single task are consistent.
- **Multilinear Factorization Machine (MFM)** is the proposed model that learns the predictive multilinear structure. We compare three variations of MFM for studying the effects of each component in the model. Forbenius norm regularizers are used as the default for all the parameters to avoid overfitting, if not specified. **MFM-F** and **MFM-F-S** both denote the variations that use the proposed predictive model in Equation 4.15, while **MFM-F-S** uses $\ell_{2,1}$ norm regularization on $\mathbf{U}$ for joint feature selection. **MFM-T** denotes the variation that uses only the tensor inner product as the predictive function, *i.e.*, the task-specific weight vector $\mathbf{u}_t$ is always set to be zeros. For all three MFM methods, we use squared loss for regression tasks and logistic loss for classification tasks.

4.4.3 Model Construction and Evaluation

For each dataset we randomly select $n\%$, 10%, and 40% of labeled samples for each task as training set, validation set, and testing set, respectively, where n is varying in the range $[10, 30]$ with the increment of 10. Validation sets are used for hyperparameter tuning for each model. Validation and testing sets do not overlap with any other. For all the methods, the dimension of latent factors $R = 20$, the learning rate $\eta = 0.1$, the initialization $\sigma = 1$, the maximum number of iterations are all set as 200. We apply grid searching to identify optimal values for each regularization hyperparameter from $\{10^{-5},\ 10^{-4},\ \cdots,\ 10^{5}\}$ for all the comparison methods.

To investigate the performance of comparison methods, we adopt accuracy (ACC), F1-score, and the area under the receiver-operator characteristic curve (AUC) on the test data as the evaluation metrics (Cao et al., 2016; He and Lawrence, 2011; Jin et al., 2013). Overall accuracy, F1-score and AUC are averaged over all tasks. The larger value of each metric indicates the better performance. For regression tasks, we adopt the mean absolute error (MAE) and root mean squared error (RMSE) on the test data as the evaluation metrics (Rendle, 2012). Overall MAE and RMSE are the averaged over all tasks. The smaller value of each metric indicates the better performance. Each experiment was repeated for 10 times, and the mean and standard deviation of each metric in each data set were reported. We conducted all the experiments on machines with 6-Core 2.4 GHz Intel Xeon CPUs and 64 GB memory.

4.4.4 Classification Tasks

Table XIII and Table XIV show the performance of all the comparison methods on the FOX and DBLP datasets. From these results, we have the several observations. First, most of the

TABLE XIII

Performance comparison on FOX dataset. The best two results are listed in bold.

Training Ratio	Measure	rMTFL	FM	TF	IteM^2	CSL-MTMV	MFM-T	MFM-F	MFM-F-S
10%	ACC	0.8816±0.011	0.7883±0.011	0.8460±0.035	0.4052±0.076	0.8986±0.011	0.9259±0.019	**0.9343±0.012**	**0.9364±0.011**
	F1	0.6911±0.035	0.2930±0.046	0.6362±0.044	0.3598±0.030	0.7335±0.029	0.7799±0.053	**0.8076±0.038**	**0.8119±0.027**
	AUC	0.9109±0.013	0.7764±0.018	0.8681±0.038	0.5326±0.036	0.9342±0.011	0.9678±0.015	**0.9763±0.008**	**0.9777±0.009**
20%	ACC	0.9039±0.013	0.8087±0.011	0.8546±0.025	0.5091±0.078	0.9264±0.005	0.9551±0.005	**0.9569±0.010**	**0.9612±0.005**
	F1	0.7654±0.026	0.3764±0.050	0.6632±0.051	0.3306±0.068	0.8004±0.012	0.8721±0.012	**0.8769±0.027**	**0.8882±0.014**
	AUC	0.9353±0.016	0.8260±0.012	0.8751±0.029	0.4954±0.043	0.9705±0.003	0.9883±0.003	**0.9885±0.006**	**0.9922±0.002**
30%	ACC	0.9314±0.005	0.8255±0.007	0.8767±0.082	0.4289±0.134	0.9390±0.004	0.9641±0.007	**0.9709±0.003**	**0.9697±0.004**
	F1	0.8051±0.015	0.4448±0.026	0.7302±0.132	0.3314±0.056	0.8341±0.012	0.9000±0.018	**0.9185±0.010**	**0.9149±0.010**
	AUC	0.9709±0.005	0.8393±0.012	0.9010±0.091	0.5365±0.039	0.9812±0.003	0.9916±0.003	**0.9949±0.001**	**0.9949±0.001**

TABLE XIV

Performance comparison on DBLP dataset. The best two results are listed in bold.

Training Ratio	Measure	rMTFL	FM	TF	IteM^2	CSL-MTMV	MFM-T	MFM-F	MFM-F-S
10%	ACC	0.8057±0.004	0.7264±0.004	0.7471±0.011	0.6223±0.004	0.7290±0.005	0.8008±0.004	**0.8058±0.004**	**0.8062±0.005**
	F1	0.5395±0.015	0.0732±0.019	0.5606±0.011	0.3176±0.007	0.4402±0.004	0.5278±0.018	**0.5469±0.014**	**0.5471±0.015**
	AUC	0.7888±0.007	0.6264±0.023	0.7723±0.009	0.5310±0.007	0.6890±0.006	0.8039±0.010	**0.8113±0.010**	**0.8120±0.009**
20%	ACC	0.8319±0.004	0.7628±0.007	0.7878±0.007	0.6309±0.003	0.7760±0.002	0.8346±0.004	**0.8374±0.004**	**0.8371±0.004**
	F1	0.6447±0.008	0.2680±0.038	0.6247±0.014	0.3494±0.006	0.5295±0.007	0.6274±0.013	**0.6499±0.012**	**0.6508±0.012**
	AUC	0.8374±0.005	0.7548±0.022	0.8200±0.010	0.5550±0.006	0.7655±0.005	0.8531±0.006	**0.8658±0.005**	**0.8632±0.005**
30%	ACC	0.8412±0.004	0.7978±0.005	0.8191±0.008	0.6256±0.003	0.8037±0.003	0.8501±0.004	**0.8527±0.004**	**0.8535±0.004**
	F1	0.6796±0.010	0.4312±0.021	0.6670±0.021	0.3569±0.009	0.5869±0.007	0.6800±0.013	**0.6891±0.012**	**0.6892±0.009**
	AUC	0.8590±0.005	0.8351±0.010	0.8498±0.009	0.5563±0.006	0.8083±0.006	0.8757±0.005	**0.8866±0.006**	**0.8866±0.006**

methods achieve better performance when the training size increases, and the proposed MFM methods consistently outperform all the other methods on both datasets. This is mainly because MFM can effectively learn the predictive multilinear structure from the full-order interactions of MTMV data. Moreover, by combining the task-specific feature map with the task-view shared multilinear feature map, MFM-F and MFM-F-S can further improve the performance. Further, it can be found that MFM-F-S almost always perform better than MFM-F, which empirically shows the effectiveness of learning features with sparse constraints.

Besides, among all the factorization based methods (*i.e.*, FM, TF and MFM), FM performs the worst. Such poor performance indicates that FM cannot discriminate important features (*e.g.*, the task indicators) for different tasks. Because all the pairwise interactions between all the features are considered in FM, the interactions between the task indicators and important features are buried by many redundant intra-view feature interactions. The ability to distinguish different tasks is critical for multi-task classification problems, since the labels of a classification task are usually dissimilar or even opposite to other classification tasks. On the other hand, by fusing multi-view information into tensor structures, the tensor-based methods can achieve relatively better performance. In addition, we find that TF performs much worse than the MFM-T. This confirms that learning only from the highest-order interactions is limited and incorporating the lower-order interactions in the predictive models can help provide more information (Cao et al., 2016; Rendle, 2010).

In addition, the rMTFL method, which learns important features for all the tasks but does not distinguish features from different views, can achieve comparative or even better performance than the state-of-the-art MTMV learning methods, *i.e.*, CSL-MTMV. This is mainly because CSL-MTMV enforces the prediction results of each view to be consistent with each other, while the text view and the image view (or the linkage view) are not similar. In contrast, the proposed MFM methods can achieve better performance by exploring the complementary information of multiple views.

TABLE XV

Performance comparison on MovieLens dataset. The best two results are listed in bold.

Training Ratio	Measure	rMTFL	FM	TF	CSL-MTMV	MFM-T	MFM-F	MFM-F-S
10%	RMSE	1.1861±0.008	1.0251±0.003	1.5679±0.099	1.05013±0.005	1.0078±0.005	**1.0069±0.005**	**0.9976±0.004**
	MAE	0.8516±0.004	0.8422±0.004	1.2497±0.088	0.8516±0.004	0.8142±0.005	**0.8082±0.005**	**0.8022±0.004**
20%	RMSE	1.0631±0.005	0.9898±0.003	1.2519±0.069	1.0214±0.004	**0.9877±0.003**	0.9977±0.003	**0.9857±0.003**
	MAE	0.8539±0.005	0.7997±0.004	0.9801±0.053	0.8294±0.004	**0.7987±0.003**	0.8023±0.003	**0.7927±0.004**
30%	RMSE	0.9917±0.003	**0.9765±0.003**	1.2066±0.061	1.0082±0.003	0.9795±0.003	0.9887±0.004	**0.9785±0.003**
	MAE	0.8159±0.003	**0.7815±0.003**	0.9380±0.045	0.8189±0.003	0.7885±0.002	0.7823±0.004	**0.7789±0.004**

TABLE XVI

Performance comparison on Amazon dataset. The best two results are listed in bold. Due to the memory overhead, rMTFL and CSL-MTMV are not compared.

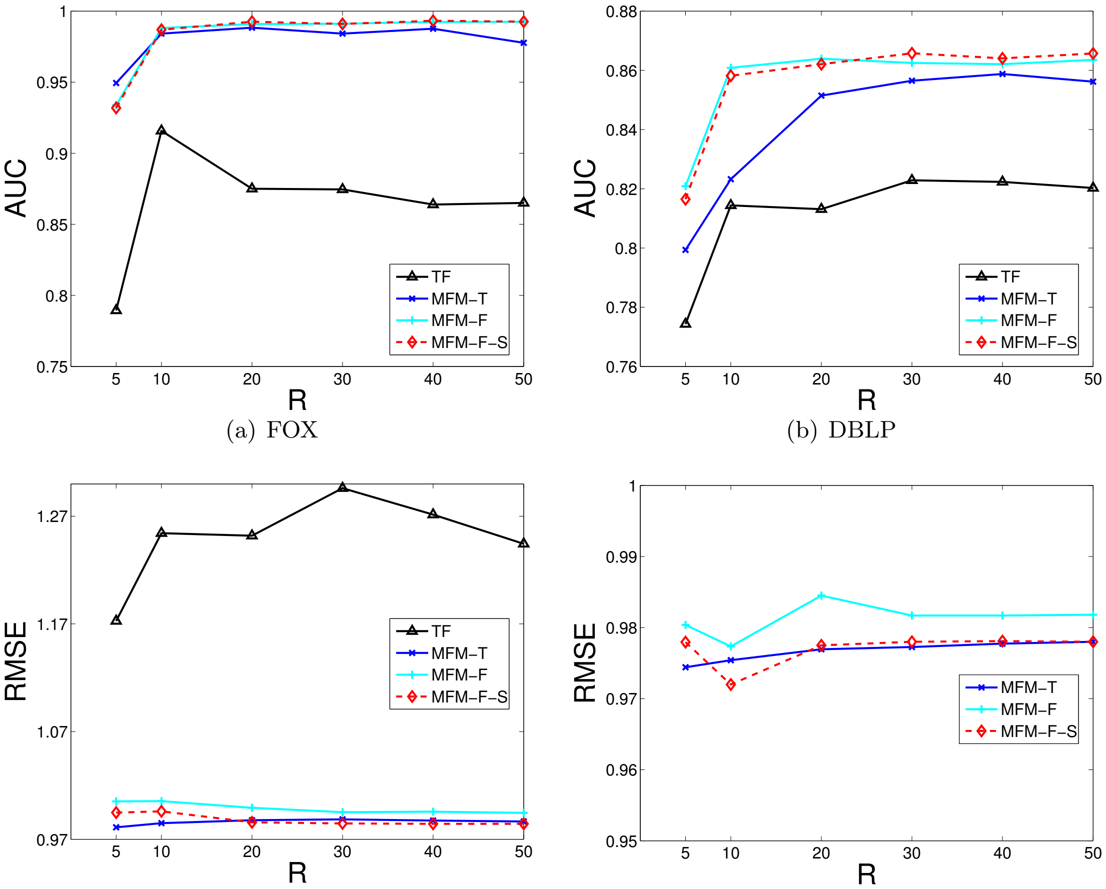
Training Ratio	Measure	FM	TF	MFM-T	MFM-F	MFM-F-S
10%	RMSE	0.9834±0.001	3.6044±0.003	**0.9775±0.001**	0.9857±0.001	**0.9825±0.002**
	MAE	0.7420±0.001	3.4574±0.005	0.7249±0.001	**0.7158±0.002**	**0.7129±0.001**
20%	RMSE	0.9814±0.001	3.5611±0.018	**0.9764±0.001**	0.9845±0.001	**0.9775±0.001**
	MAE	0.7343±0.002	3.3965±0.030	0.7255±0.001	**0.7112±0.001**	**0.7086±0.001**
30%	RMSE	0.9782±0.002	3.4962±0.018	**0.9705±0.002**	0.9841±0.001	**0.9733±0.001**
	MAE	0.7257±0.002	3.2945±0.034	**0.7001±0.001**	0.7115±0.001	**0.7078±0.001**

4.4.5 Regression Tasks

Table XV and Table XVI report the performance comparison on the MovieLens and Amazon datasets. Note that IteM^2 is not compared since it can only work for classification tasks. Besides, due to the high memory complexity CSL-MTMV and rMTFL cannot be applied to the large-scale Amazon dataset. From these results, we can observe that the proposed MFM methods outperform most of the comparison methods on both datasets, especially when the training data is limited. This is because when less instances are available, some users and items may hardly appear during training, making it harder to learn the model parameters for the user view and the item view. By incorporating the bias terms for each view in the full-order tensor, the proposed MFM models can explore the information from other complementary views (*e.g.*, the tag view or the text view) to alleviate the issue and to improve the performance. We also notice that FM achieves much more competitive performance in regression tasks than in classification tasks. This is due to the fact that the regression tasks in the experiments are more similar and thus it is less important to learn task-specific features. Furthermore, we observe that the performance of TF on the Amazon dataset is unacceptably low, which is due to the fact that some instances do not have any text features. Because TF can only learn from the highest-order interactions, its performance is significantly impacted by the existence of the partially missing view. In contrast, by taking the full-order interactions into consideration, MFM methods can easily deal with the partially missing view.

4.4.6 Hyperparameter Analysis

The number of latent factors R is an important hyperparameter for the proposed MFM methods involving the CP factorization. We analyze different values of R and report the results in Figure 10. Due to space limit, we only report AUC for classification tasks and RMSE for regression tasks using 20% of instances as training set. Compared with the other tensor-based method (TF), the performance of MFM methods is much more stable when $R \geq 20$ for all four datasets. It validates the importance of including lower-order interactions in the predictive model.



(a) FOX

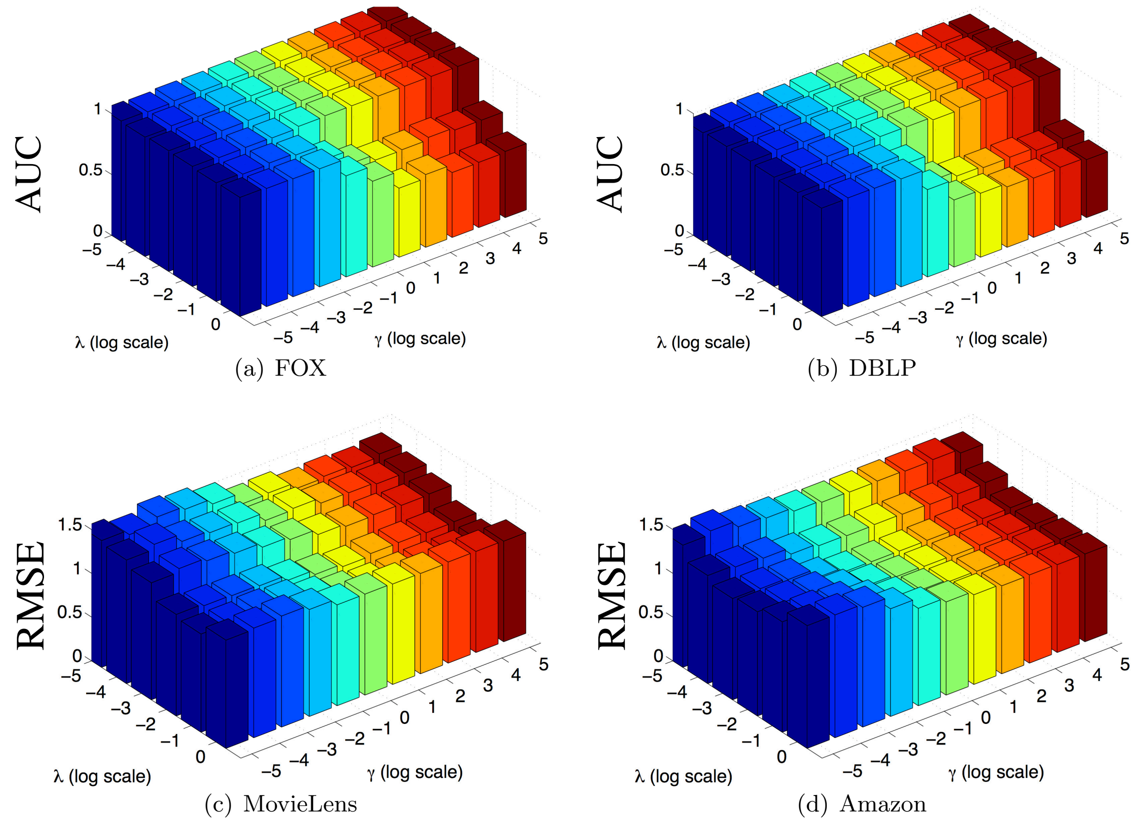
(b) DBLP

(c) MovieLens

(d) Amazon

Figure 10. Sensitivity analysis of number of latent factor.

To explore the effects of the regularization hyperparameters on the performance, we run with different values for λ and γ. Since there are no much differences between the results of MFM-F-S and those of MFM-F, we only report the results using MFM-F-S. From 11(a) and 11(b), we can clearly see that the performance of classification is fairly stable for most cases. However, when λ and γ are both large, the model parameters can be very small, which



(a) FOX

(b) DBLP

(c) MovieLens

(d) Amazon

Figure 11. Sensitivity analysis of regularization hyperparameters.

decreases the value of AUC. From 11(c) and 11(d), we can observe that for regression tasks, the RMSE is much lower when the value of γ is large and the value of λ is in the range from 10^{-3} to 10^{-1}. This could indicate that the task-specific weight vector $\mathbf{u}_t$ is less important for the prediction. This confirms our observation that the tasks in the regression problem are much more similar than in the classification problem.

4.5 Related Work

Multi-Task Learning: MTL aims to improve the performance of related tasks by learning a model which is able to capture intrinsic relatedness across tasks (Caruana, 1998). The current work mainly focuses on joint feature selection and feature learning, where the task relatedness is explicitly expressed as a shared part of the tasks (Ando and Zhang, 2005; Argyriou et al., 2008; Evgeniou and Pontil, 2007; Evgeniou and Pontil, 2004; Obozinski et al., 2010). Most of these methods make the assumption that different tasks share a common representation/structure. However, in practical applications it is too restrictive to constrain all tasks to share a common set of features, due to the heterogeneity of tasks. Some recent methods proposed to capture different types of relationships using a composite regularization (Chen et al., 2011; Gong et al., 2012). In particular, the alternating structure optimization (ASO) algorithm (Ando and Zhang, 2005) and its convex version cASO algorithm (Chen et al., 2009) decompose the predictive model into the task-specific and task-shared feature mapping, which can be cast as special cases of our framework when $V = 1$.

Multi-View Learning: MVL concerns about exploiting different views on the same object to make a more accurate learning. In essence, MVL explores diverse representations from experiential inputs such that a variety of problems could be formulated and solved (Ceci et al., 2012). In this fashion, the strengths of each view are amplified and the weaknesses are alleviated. There are currently a plethora of studies available for MVL. Interested readers are referred to (Xu et al., 2013) for a comprehensive survey of these techniques and applications. The most related work to ours is that of (Cao et al., 2014a; Cao et al., 2016) who introduced and explored the tensor product operator to integrate different views together in a joint tensor. The advantage of tensor representation is that it enables not only to record the information from multiple views, but also strengthen and capture the relationships between them (Li et al., 2016). In addition, various tensor factorizations (Kolda and Bader, 2009) can be modelled to learn dependencies between variables, each of which imply different hypotheses about the data. In this study, we employ the CP factorization to facilitate the learning process, but it can be easily extended also to other types of factorizations.

Multi-Task Multi-View Learning: The IteM^2 algorithm (He and Lawrence, 2011) was first proposed by He *et al.* for MTMV learning. Since IteM^2 is a transductive learning method, prediction cannot be made to independent, or unknown testing data samples. Besides, it can only deal with classification problems with nonnegative feature values. Assuming the predictive models should be consistent among different views, co-regularization based methods were later developed (Jin et al., 2013; Zhang and Huan, 2012). Specifically, Zhang *et al.* proposed regMVMT algorithm (Zhang and Huan, 2012) that minimizes the difference of the predictive models for different tasks on the same view; Jin *et al.* proposed a more generalized algorithm, CSL-MTMV (Jin et al., 2013), which assumes a low-dimensional subspace is shared among multiple related tasks that have common views. These methods assume that all the views are similar to each other, while such assumption may not be appropriate especially for heterogeneous data. It also makes these methods have difficulty in learning from partially observed views.

CHAPTER 5

MODELING MULTI-VIEW RELATIONAL DATA

(This chapter was previously published as “Learning from Multi-View Multi-Way Data via Structural Factorization Machines”, in WWW’18 (Lu et al., 2018). DOI: http://dx.doi.org/10.1145/3178876.3186071.)

5.1 Introduction

With the ability to access massive amounts of heterogeneous data from multiple sources, multi-view data have become prevalent in many real-world applications. For instance, in recommender systems, online review sites (like Amazon and Yelp) have access to contextual information of shopping histories of users, the reviews written by the users, the categorizations of the items, as well as the friends of the users. Each view may exhibit pairwise relations (e.g., the friendships between users) or even higher-order relations (e.g., a customer write a review for a product) among entities (such as customers, products, and reviews), and can be represented in a multi-way relational data structure, i.e., tensor. Since different views usually provide complementary information (Cao et al., 2014a; Cao et al., 2016; Lu et al., 2017), how to effectively incorporate information from multiple structural views is critical to good prediction performance for various machine learning tasks.

Typically, a predictive model is defined as a function of predictor variables (e.g., the customer id, the product id, and the categories of the product) to some target (e.g., the rating). The most common approach in predictive modeling for multi-view relational data is to describe samples with feature vectors that are flattened and concatenated from structural views, and apply a classical vector-based method, such as linear regression (LR) and support vector machines (SVMs), to learn the target function from observed samples. Recent works have shown that linear models fail for tasks with very sparse data (Rendle, 2012). A variety of methods have been proposed to address the data sparsity issue by factorizing the monomials (or feature interactions) with kernels, such as the ANOVA kernels used in FMs (Rendle, 2012; Blondel et al., 2016a) and polynominal kernels used in polynominal networks (Livni et al., 2014; Blondel et al., 2016b). However, the disadvantages of this approach are that (1) the important structural information of each view will be discarded which may lead to the degraded prediction performance and (2) the feature vectors can be extremely large such that learning and prediction would be very slow or infeasible, especially if each view involves relations of high cardinality. For example, including the relation "friends of a user" in the feature vector (represented by their IDs) can result in a very long feature vector. Further, it will repeatedly appear in many samples that involve the given user.

Matrix/tensor factorization models have been a topic of interest in the areas of multi-way data analysis, e.g., community detection (He et al., 2016), collaborative filtering (Koren, 2008; Rendle and Schmidt-Thieme, 2010), knowledge graph completion (Zhang et al., 2017), and neuroimage analysis (He et al., 2014). Assuming multi-view data have the same underlying low-rank structure (at least in one mode), coupled data analysis such as collective matrix factorization (CMF) (Singh and Gordon, 2008) and coupled matrix and tensor factorization

(CMTF) (Acar et al., 2011) that jointly factorize multiple matrices (or tensors) has been applied to applications such as clustering and missing data recovery. However, they are only applicable to categorical variables. Moreover, since existing coupled factorization models are unsupervised, the importance of each structural view in modeling the target value cannot be automatically learned. Furthermore, when applying these models to data with rich meta information (e.g., friendships) but extremely sparse target values (e.g., ratings), it is very likely the learning process will be dominated by the meta information without manual tuning some hyperparameters, e.g., the weights of the fitting error of each matrix/tensor in the objective function (Singh and Gordon, 2008), the weights of different types of latent factors in the predictive models (Koren, 2010), or the regularization hyperparamters of latent factor alignment (Lu et al., 2016).

In this Chapter, we propose a general and flexible framework for learning the predictive structure from the complex relationships within the multi-view relational data. Each view of an instance in this framework is represented by a tensor that describes the multi-way relations of subsets of entities, and different views have some entities in common. Constructing the tensors for each instance may not be realistic for real-world applications in terms of space and computational complexity, and the model parameters can have exponential growth and tend to be overfitting. In order to preserve the structural information of multi-view data without physically constructing the tensors, we introduce structural factorization machines (SFMs) that can learn the consistent representations in the latent feature spaces shared in the multi-view tensors while automatically adjust the contribution of each view in the predictive model. Furthermore, we provide an efficient method to avoid redundant computing on repeating patterns stemming from the relational structure of the data, such that SFMs can make the same predictions but with largely speed up computation.

The contributions of this Chapter are summarized as follows:

- We introduce a novel multi-tensor framework for mining data from heterogeneous domains, which can explore the high order correlations underlying multi-view relational data in a generic predictive model.
- We develop structural factorization machines (SFMs) tailored for learning the common latent spaces shared in multi-view tensors and automatically adjusting the importance of each view in the predictive model. The complexity of SFMs is linear in the number of features, which makes SFMs suitable to large-scale problems.
- Extensive experiments on eight real-world datasets are performed along with comparisons to existing state-of-the-art factorization models to demonstrate its advantages.

5.2 Problem Formulation

Our problem is different from conventional multi-view learning approaches where multiple views of data are assumed independent and disjoint, and each view is described by a vector. We formulate the multi-view learning problem using coupled analysis of multi-view features in the form of multiple tensors.

Suppose that the problem includes V views where each view consists of a collection of subsets of entities (such as person, company, location, product) and different views have some entities

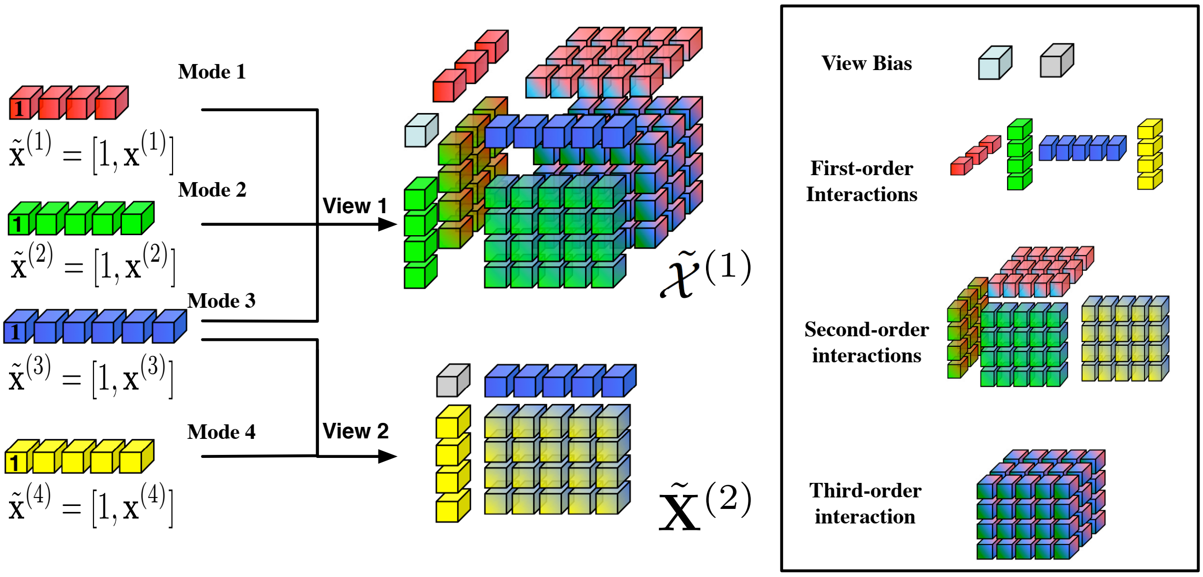

Mode 1
$\tilde{x}^{(1)} = [1, x^{(1)}]$
Mode 2
$\tilde{x}^{(2)} = [1, x^{(2)}]$
Mode 3
$\tilde{x}^{(3)} = [1, x^{(3)}]$
Mode 4
$\tilde{x}^{(4)} = [1, x^{(4)}]$
1
1
1
1
View 1
View 2
$\tilde{\mathcal{X}}^{(1)}$
$\tilde{\mathbf{X}}^{(2)}$
View Bias
First-order Interactions
Second-order interactions
Third-order interaction

Figure 12. Example of multiple structural views, where $\tilde{\mathcal{X}}^{(1)} = \tilde{\mathbf{x}}^{(1)} \circ \tilde{\mathbf{x}}^{(2)} \circ \tilde{\mathbf{x}}^{(3)}$ and $\tilde{\mathbf{X}}^{(2)} = \tilde{\mathbf{x}}^{(3)} \circ \tilde{\mathbf{x}}^{(4)}$.

in common. We denote a view as a tuple $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(M)}), M \geq 2$, where $\mathbf{x}^{(m)} \in \mathbb{R}^{I_m}$ is a feature vector associated with the entity m. Inspired by (Cao et al., 2016), we construct tensor representation for each view over its entities by

$$\tilde{\mathcal{X}} = \tilde{\mathbf{x}}^{(1)} \circ \tilde{\mathbf{x}}^{(2)} \circ \cdots \circ \tilde{\mathbf{x}}^{(M)} \in \mathbb{R}^{(1+I_1)\times\cdots\times(1+I_M)},$$

where $\tilde{\mathbf{x}}^{(m)} = [1; \mathbf{x}^{(m)}] \in \mathbb{R}^{1+I_m}$ and $\circ$ is the outer product operator. In this manner, the full-order interactions between entities are embedded within the tensor structure, which not only provides a unified and compact representation for each view, but also facilitate efficient design methods. Figure 12 shows an example of two structural views, where the first view consists of the full-order interactions among the first three modes (e.g., review text, item ID, and user ID), and the second view consists of the full-order interactions among the last two modes (e.g., user ID and friend IDs).

After generating the tensor representation for each view, we define the multi-view learning problem as follows. Given a training set $\mathfrak{D} = \big\{\big(\{\tilde{\mathcal{X}}_n^{(1)}, \tilde{\mathcal{X}}_n^{(2)}, \cdots, \tilde{\mathcal{X}}_n^{(V)}\},\ y_n\big) \mid n \in [1:N]\big\}$, where $\tilde{\mathcal{X}}_n^{(v)} \in \mathbb{R}^{(1+I_1)\times\cdots\times(1+I_{M_v})}$ is the tensor representation in the v-th view for the n-th instance, y_n is the response of the n-th instance, M_v is the number of the constitutive modes in the v-th view, and N is the number of labeled instances. We assume different views have common entities, thus the resulting tensors will share common modes, e.g., the third mode in Fig Figure 12. As we are concerned with predicting unknown values of multiple coupled tensors, our goal is to leverage the relational information from all the views to help predict the unlabeled instances, as well as to use the complementary information among different views to improve the performance. Specifically, we are interested in finding a predictive function $f : \mathfrak{X}^{(1)} \times \mathfrak{X}^{(2)} \cdots \times \mathfrak{X}^{(V)} \rightarrow \mathfrak{Y}$ that minimizes the expected loss, where $\mathfrak{X}^{(v)}, v \in [1:V]$ is the input space in the v-th view and $\mathfrak{Y}$ is the output space.

5.3 Methodology

In this section, we first discuss how to design the predictive models for learning from multiple coupled tensors. We then derive structural factorization machines (SFMs) that can learn the common latent spaces shared in multi-view coupled tensors and automatically adjust the importance of each view in the predictive model.

5.3.1 Predictive Models

Without loss of generality, we take two views as an example to introduce our basic design of the predictive models. Specifically, we consider coupled analysis of a third-order tensor and a matrix with one mode in common, as shown in Figure 12. Given an input instance $\big(\{\tilde{\mathcal{X}}^{(1)}, \tilde{\mathbf{X}}^{(2)}\},\ y\big)$, where $\tilde{\mathcal{X}}^{(1)} = \tilde{\mathbf{x}}^{(1)} \circ \tilde{\mathbf{x}}^{(2)} \circ \tilde{\mathbf{x}}^{(3)} \in \mathbb{R}^{(1+I)\times(1+J)\times(1+K)}$ and $\tilde{\mathbf{X}}^{(2)} = \tilde{\mathbf{x}}^{(3)} \circ \tilde{\mathbf{x}}^{(4)} \in \mathbb{R}^{(1+K)\times(1+L)}$. An intuitive solution is to build the following multiple linear model:

$$f\left(\left\{\tilde{\mathcal{X}}^{(1)}, \tilde{\mathbf{X}}^{(2)}\right\}\right) = \left\langle \tilde{\mathcal{W}}^{(1)}, \tilde{\mathcal{X}}^{(1)} \right\rangle + \left\langle \tilde{\mathbf{W}}^{(2)}, \tilde{\mathbf{X}}^{(2)} \right\rangle \tag{5.1}$$

where $\tilde{\mathcal{W}}^{(1)} \in \mathbb{R}^{(1+I)\times(1+J)\times(1+K)}$ and $\tilde{\mathbf{W}}^{(2)} \in \mathbb{R}^{(1+K)\times(1+L)}$ are the weights for each view to be learned.

However, in this case it does not take into account the relations and differences between two views. In order to incorporate the relations between two views and also discriminate the importance of each view, we introduce an indicator vector $\mathbf{e}_v \in \mathbb{R}^V$ for each view v as

$$\mathbf{e}_v = [\underbrace{0, \cdots, 0}_{\text{v-1}}, 1, 0, \cdots, 0]^{\mathrm{T}},$$

and transform the predictive model in Equation 5.1 into

$$f\left(\left\{\tilde{\mathcal{X}}^{(1)}, \tilde{\mathbf{X}}^{(2)}\right\}\right) = \left\langle \hat{\mathcal{W}}^{(1)}, \tilde{\mathcal{X}}^{(1)} \circ \mathbf{e}_1 \right\rangle + \left\langle \hat{\mathcal{W}}^{(2)}, \tilde{\mathbf{X}}^{(2)} \circ \mathbf{e}_2 \right\rangle, \tag{5.2}$$

where $\hat{\mathcal{W}}^{(1)} \in \mathbb{R}^{(1+I)\times(1+J)\times(1+K)\times 2}$ and $\hat{\mathcal{W}}^{(2)} \in \mathbb{R}^{(1+K)\times(1+L)\times 2}$.

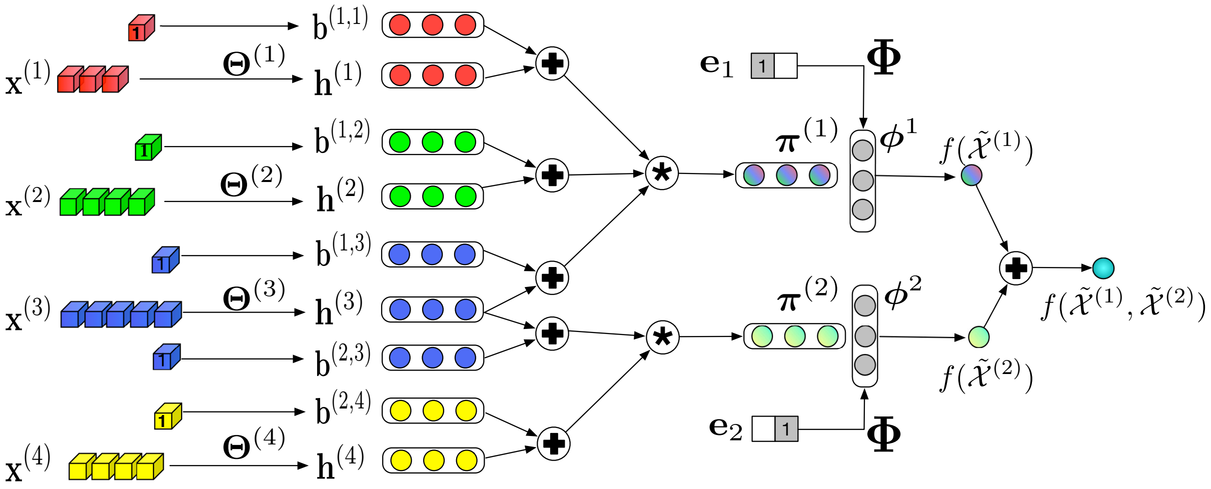


Figure 13. Example of the computational graph in a structural factorization machine, given the input $\tilde{\mathcal{X}}^{(1)}$ and $\tilde{\mathbf{X}}^{(2)}$. By jointly factorizing weight tensors, the $\mathbf{h}^{(m)}$ can be regarded as the latent representation of the feature $\mathbf{x}^{(m)}$ in m-th mode, and $\boldsymbol{\pi}^{(v)}$ can be regarded as the joint representation of all the modes in the v-th view, which can be easily computed through the Hadamard product. The contribution of $\boldsymbol{\pi}^{(v)}$ to the final prediction score is automatically adjusted by the weight vector $\boldsymbol{\phi}^{v}$.

Directly learning the weight tensors $\hat{\mathcal{W}}$s leads to two drawbacks. First, the weight parameters are learned independently for different modes and different views. When the feature interactions rarely (or even never) appear during training, it is unlikely to learn the associated parameters appropriately. Second, the number of parameters in Equation 5.2 is exponential to the number of features, which prones to overfitting and ineffective on sparse data. Here, we assume each weight tensor has a low-rank approximation, and $\hat{\mathcal{W}}^{(1)}$ and $\hat{\mathcal{W}}^{(2)}$ can be decomposed by CP factorization as

$$\begin{aligned}\hat{\mathcal{W}}^{(1)} &= [\![\hat{\mathbf{\Theta}}^{(1,1)}, \hat{\mathbf{\Theta}}^{(1,2)}, \hat{\mathbf{\Theta}}^{(1,3)}, \mathbf{\Phi}]\!] \\ &= [\![[\mathbf{b}^{(1,1)}; \mathbf{\Theta}^{(1)}], [\mathbf{b}^{(1,2)}; \mathbf{\Theta}^{(2)}], [\mathbf{b}^{(1,3)}; \mathbf{\Theta}^{(3)}], \mathbf{\Phi}]\!],\end{aligned}$$

and

$$\hat{\mathcal{W}}^{(2)} = [\![\hat{\mathbf{\Theta}}^{(2,3)}, \hat{\mathbf{\Theta}}^{(2,4)}, \mathbf{\Phi}]\!] = [\![[\mathbf{b}^{(2,3)}; \mathbf{\Theta}^{(3)}], [\mathbf{b}^{(2,4)}; \mathbf{\Theta}^{(4)}], \mathbf{\Phi}]\!],$$

where $\mathbf{\Theta}^{(m)} \in \mathbb{R}^{I_m \times R}$ is the factor matrix for the features in the m-th mode. It is worth noting that $\mathbf{\Theta}^{(3)}$ is shared in the two views. $\mathbf{\Phi} \in \mathbb{R}^{2 \times R}$ is the factor matrix for the view indicator, and $\mathbf{b}^{(v,m)} \in \mathbb{R}^{1 \times R}$, which is always associated with the constant one in $\tilde{\mathbf{x}}^{(m)} = [1; \mathbf{x}^{(m)}]$, represents the bias factors of the m-th mode in the v-th view. Through $\mathbf{b}^{(v,m)}$, the lower-order interactions (the interactions excluding the features from the m-th mode) in the v-th view are explored in the predictive function.

Then we can transform Equation 5.2 into

$$
\begin{aligned}
&\left\langle \hat{\mathcal{W}}^{(1)}, \tilde{\mathcal{X}}^{(1)} \circ \mathbf{e}_1 \right\rangle + \left\langle \hat{\mathcal{W}}^{(2)}, \tilde{\mathbf{X}}^{(2)} \circ \mathbf{e}_2 \right\rangle \\
=&\sum_{r=1}^{R} \left\langle \hat{\boldsymbol{\theta}}_r^{(1,1)} \circ \hat{\boldsymbol{\theta}}_r^{(1,2)} \circ \hat{\boldsymbol{\theta}}_r^{(1,3)} \circ \boldsymbol{\phi}_r \ , \ \tilde{\mathbf{x}}^{(1)} \circ \tilde{\mathbf{x}}^{(2)} \circ \tilde{\mathbf{x}}^{(3)} \circ \mathbf{e}_1 \right\rangle \\
&+ \sum_{r=1}^{R} \left\langle \hat{\boldsymbol{\theta}}_r^{(2,3)} \circ \hat{\boldsymbol{\theta}}_r^{(2,4)} \circ \boldsymbol{\phi}_r \ , \ \tilde{\mathbf{x}}^{(3)} \circ \tilde{\mathbf{x}}^{(4)} \circ \mathbf{e}_2 \right\rangle \\
=&\boldsymbol{\phi}^1 \left(\prod_{m=1}^{3} * \left(\tilde{\mathbf{x}}^{(m)\mathrm{T}} \hat{\boldsymbol{\Theta}}^{(1,m)} \right) \right)^{\mathrm{T}} + \boldsymbol{\phi}^2 \left(\prod_{m=3}^{4} * \left(\tilde{\mathbf{x}}^{(m)\mathrm{T}} \hat{\boldsymbol{\Theta}}^{(2,m)} \right) \right)^{\mathrm{T}} \\
=&\boldsymbol{\phi}^1 \left(\prod_{m=1}^{3} * \left(\mathbf{x}^{(m)\mathrm{T}} \boldsymbol{\Theta}^{(m)} + \mathbf{b}^{(1,m)} \right) \right)^{\mathrm{T}} + \boldsymbol{\phi}^2 \left(\prod_{m=3}^{4} * \left(\mathbf{x}^{(m)\mathrm{T}} \boldsymbol{\Theta}^{(m)} + \mathbf{b}^{(2,m)} \right) \right)^{\mathrm{T}}
\end{aligned}
\tag{5.3}
$$

where $*$ is the Hadamard (elementwise) product and $\boldsymbol{\phi}^v \in \mathbb{R}^{1\times R}$ is the v-th row of the factor matrix $\boldsymbol{\Phi}$.

For convenience, we let $\mathbf{h}^{(m)} = \boldsymbol{\Theta}^{(m)\mathrm{T}} \mathbf{x}^{(m)}$, $S_M(v)$ denote the set of modes in the v-th views, $\boldsymbol{\pi}^{(v)} = \prod_{m\in S_M(v)} * \left(\mathbf{h}^{(m)} + \mathbf{b}^{(v,m)\mathrm{T}} \right)$, and $\boldsymbol{\pi}^{(v,-m)} = \prod_{m'\in S_M(v), m'\neq m} * \left(\mathbf{h}^{(m')} + \mathbf{b}^{(v,m')\mathrm{T}} \right)$. The predictive model for the general cases is given as follows

$$
\begin{aligned}
f(\{\tilde{\mathcal{X}}^{(v)}\}) &= \sum_{v=1}^{V} \left\langle \hat{\mathcal{W}}^{(v)}, \tilde{\mathcal{X}}^{(v)} \circ \mathbf{e}_v \right\rangle \\
&= \sum_{v=1}^{V} \boldsymbol{\phi}^v \prod_{m\in S_M(v)} * \left(\mathbf{x}^{(m)\mathrm{T}} \boldsymbol{\Theta}^{(m)} + \mathbf{b}^{(v,m)} \right)^{\mathrm{T}} \\
&= \sum_{v=1}^{V} \boldsymbol{\phi}^v \prod_{m\in S_M(v)} * \left(\mathbf{h}^{(m)} + \mathbf{b}^{(v,m)\mathrm{T}} \right)
\end{aligned}
\tag{5.4}
$$

A graphical illustration of the proposed model is shown in Figure 13. We name this model as structural factorization machines (SFMs). Clearly, the parameters are jointly factorized, which benefits parameter estimation under sparsity since dependencies exist when the interactions share the same features. Therefore, the model parameters can be effectively learned without direct observations of such interactions especially in highly sparse data. More importantly, after factorizing the weight tensor $\hat{\mathcal{W}}$s, there is no need to construct the input tensor physically. Furthermore, the model complexity is linear in the number of original features. In particular, the model complexity is $O(R(V + I + \sum_v M_v))$, where M_v is the number of modes in the v-th view.

5.3.2 Learning Structural Factorization Machines

Following the traditional supervised learning framework, we propose to learn the model parameters by minimizing the following regularized empirical risk:

$$\mathcal{R} = \frac{1}{N}\sum_{n=1}^{N} \ell\left(f(\{\mathcal{X}_n^{(v)}\}), y_n\right) + \lambda\Omega(\mathbf{\Phi}, \{\mathbf{\Theta}^{(m)}\}, \{\mathbf{b}^{(v,m)}\}) \tag{5.5}$$

where ℓ is a prescribed loss function, Ω is the regularizer encoding the prior knowledge of $\{\mathbf{\Theta}^{(m)}\}$ and $\mathbf{\Phi}$, and $\lambda \geq 0$ is the regularization parameter that controls the trade-off between the empirical loss and the prior knowledge.

The partial derivative of $\mathcal{R}$ w.r.t. $\mathbf{\Theta}^{(m)}$ is given by

$$\frac{\partial \mathcal{R}}{\partial \mathbf{\Theta}^{(m)}} = \frac{\partial \mathcal{L}}{\partial f}\frac{\partial f}{\partial \mathbf{\Theta}^{(m)}} + \lambda\frac{\partial \Omega_\lambda(\mathbf{\Theta}^{(m)})}{\partial \mathbf{\Theta}^{(m)}} \tag{5.6}$$

where $\frac{\partial \mathcal{L}}{\partial f} = \frac{1}{N}\left[\begin{array}{c} \frac{\partial \ell_1}{\partial f}, \cdots, \frac{\partial \ell_N}{\partial f} \end{array}\right]^{\mathrm{T}} \in \mathbb{R}^N$.

For convenience, we let $S_V(m)$ denote the set of views that contains the m-th mode, $\mathbf{X}^{(m)} = [\mathbf{x}_1^{(m)}, \cdots, \mathbf{x}_N^{(m)}]$, $\mathbf{\Pi}^{(v)} = [\boldsymbol{\pi}_1^{(v)}, \cdots, \boldsymbol{\pi}_N^{(v)}]^{\mathrm{T}}$ and $\mathbf{\Pi}^{(v,-m)} = [\boldsymbol{\pi}_1^{(v,-m)}, \cdots, \boldsymbol{\pi}_N^{(v,-m)}]^{\mathrm{T}}$. We then have that

$$\frac{\partial \mathcal{L}}{\partial f}\frac{\partial f}{\partial \mathbf{\Theta}^{(m)}} = \mathbf{X}^{(m)}\left(\sum_{v\in S_V(m)}\left(\left(\frac{\partial \mathcal{L}}{\partial f}\boldsymbol{\phi}^{v}\right) * \mathbf{\Pi}^{(v,-m)}\right)\right) \tag{5.7}$$

Similarly, the partial derivative of $\mathcal{R}$ w.r.t. $\mathbf{b}^{(v,m)}$ is given by

$$\begin{aligned}\frac{\partial \mathcal{R}}{\partial \mathbf{b}^{(v,m)}} &= \frac{\partial \mathcal{L}}{\partial f}\frac{\partial f}{\partial \mathbf{b}^{(v,m)}} + \lambda\frac{\partial \Omega_\lambda(\mathbf{b}^{(v,m)})}{\partial \mathbf{b}^{(v,m)}} \\ &= \mathbf{1}^{\mathrm{T}}\left(\left(\frac{\partial \mathcal{L}}{\partial f}\boldsymbol{\phi}^{v}\right) * \mathbf{\Pi}^{(v,-m)}\right) + \lambda\frac{\partial \Omega_\lambda(\mathbf{b}^{(v,m)})}{\partial \mathbf{b}^{(v,m)}}\end{aligned} \tag{5.8}$$

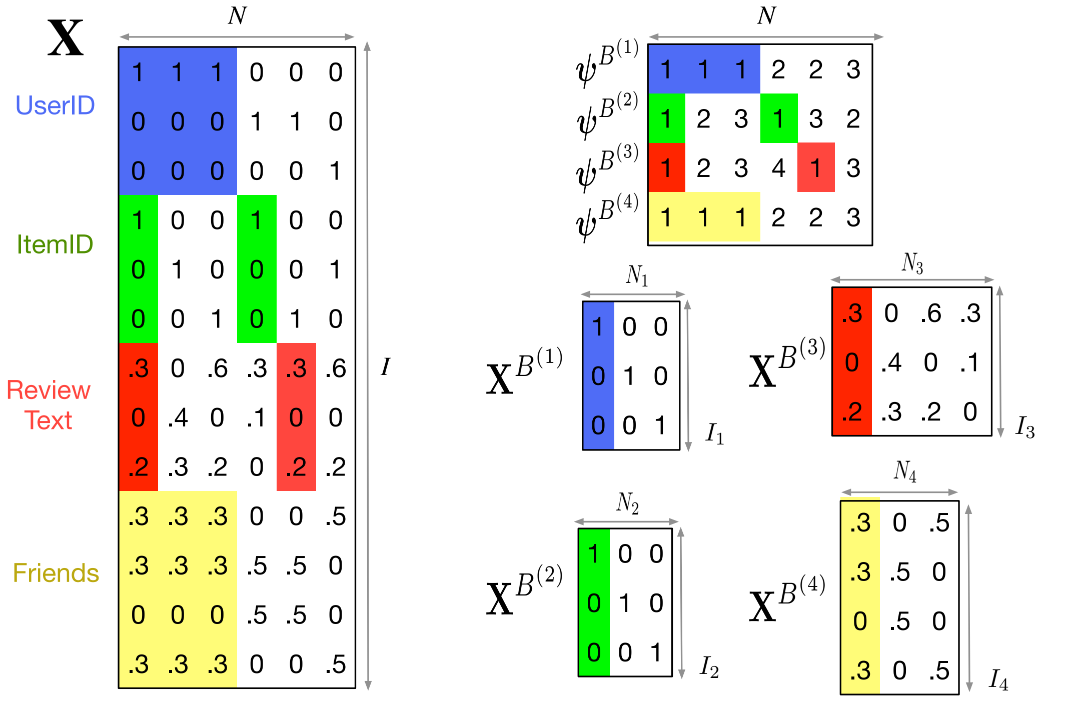
The partial derivative of $\mathcal{R}$ w.r.t. $\mathbf{\Phi}$ is given by

$$\frac{\partial \mathcal{R}}{\partial \mathbf{\Phi}} = \left[\ \left(\tfrac{\partial \mathcal{L}}{\partial f}\right)^{\mathrm{T}}\mathbf{\Pi}^{(1)}\ ;\ \cdots\ ;\ \left(\tfrac{\partial \mathcal{L}}{\partial f}\right)^{\mathrm{T}}\mathbf{\Pi}^{(V)}\ \right] + \lambda\frac{\partial \Omega_\lambda(\mathbf{\Phi})}{\partial \mathbf{\Phi}} \tag{5.9}$$

Finally, the gradient of $\mathcal{R}$ can be formed by vectorizing the partial derivatives with respect to each factor matrix and concatenating them all, i.e.,

$$\nabla\mathcal{R} = \begin{bmatrix} \mathrm{vec}(\frac{\partial\mathcal{R}}{\partial\mathbf{\Theta}^{(1)}}) \\ \vdots \\ \mathrm{vec}(\frac{\partial\mathcal{R}}{\partial\mathbf{\Theta}^{(M)}}) \\ \mathrm{vec}(\frac{\partial\mathcal{R}}{\partial\mathbf{b}^{(1,1)}}) \\ \vdots \\ \mathrm{vec}(\frac{\partial\mathcal{R}}{\partial\mathbf{b}^{(V,M)}}) \\ \mathrm{vec}(\frac{\partial\mathcal{R}}{\partial\mathbf{\Phi}}) \end{bmatrix} \tag{5.10}$$

Once we have the function, $\mathcal{R}$ and gradient, $\nabla\mathcal{R}$, we can use any gradient-based optimization algorithm to compute the factor matrices. For the results presented in this Chapter, we use the Adaptive Moment Estimation (Adam) optimization algorithm (Kingma and Ba, 2014) for parameter updates. Adam is an adaptive version of gradient descent that controls individual adaptive learning rates for different parameters from estimates of first and second moments of the gradient. It combines the best properties of the AdaGrad (Duchi et al., 2011), which works well with sparse gradients, and RMSProp (Hinton et al., 2012), which works well in on-line and non-stationary settings. Readers can refer to (Kingma and Ba, 2014) for details of the Adam optimization algorithm.



(a) Plain Format of Feature Matrix

(b) Relational Structure Representation

Figure 14. (a) Feature vectors of the same entity repeatedly appear in the plain formatted feature matrix $\mathbf{X}$. (b) Repeating patterns in $\mathbf{X}$ can be formalized by the relational structure $\mathbf{B}$ of each mode.

5.3.3 Efficient Computing with Relational Structures

In relational domains, we can often observe that feature vectors of the same entity repeatedly appear in the plain formatted feature matrix $\mathbf{X}$, where $\mathbf{X} = [\mathbf{X}^{(1)}; \cdots ; \mathbf{X}^{(M)}] \in \mathbb{R}^{I \times N}$ and $\mathbf{X}^{(m)} \in \mathbb{R}^{I_m \times N}$ is the feature matrix in the m-th mode. Consider Figure 14(a) as an example, where the parts highlighted in yellow in the forth mode (which represents the friends of the user) are repeatedly appear in the first three columns. Clearly, these repeating patterns stem from the relational structure of the same entity.

In the following, we show how the proposed SFM method can make use of relational structure of each mode, such that the learning and prediction can be scaled to the number of features in data involving relations of high cardinality. We adopt the idea from (Rendle, 2013) to avoid redundant computing on repeating patterns over a set of feature vectors.

Let $\mathcal{B} = \{(\mathbf{X}^{B^{(m)}}, \psi^{B^{(m)}})\}_{m=1}^{M}$ be the set of relational structures, where $\mathbf{X}^{B^{(m)}} \in \mathbb{R}^{I_m \times N_m}$ denotes the relational matrix of m-th mode, $\psi^{B^{(m)}} : \{1, \cdots, N\} \rightarrow \{1, \cdots, N_m\}$ denotes the mapping from columns in the feature matrix $\mathbf{X}$ to columns within $\mathbf{X}^{B^{(m)}}$. For the sake of simplicity, we drop the index B in ψ^B whenever the block mapping is clear. From $\mathcal{B}$, one can reconstruct $\mathbf{X}$ by concatenating the corresponding columns of the relational matrices using the mappings. For instance, the feature vector $\mathbf{x}_n$ of the n-th case in the plain feature matrix $\mathbf{X}$ is represented as $\mathbf{x}_n = [\mathbf{x}^{(1)}_{\psi(n)}; \cdots; \mathbf{x}^{(M)}_{\psi(n)}]$. Figure 14(b) shows an example how the feature matrix can be represented in relational structures. For instance, the forth column of the feature matrix $\mathbf{X}$ can be represented as $\mathbf{x}_4 = [\mathbf{x}^{(1)}_{\psi(4)}; \mathbf{x}^{(2)}_{\psi(4)}; \mathbf{x}^{(3)}_{\psi(4)}; \mathbf{x}^{(4)}_{\psi(4)}] = [\mathbf{x}^{B^{(1)}}_2; \mathbf{x}^{B^{(2)}}_1; \mathbf{x}^{B^{(3)}}_4; \mathbf{x}^{B^{(4)}}_2]$.

Let $N_z(\mathbf{A})$ denote the number of non-zeros in a matrix $\mathbf{A}$. The space required for using relational structures to represent the input data is $|\mathcal{B}| = NM + \sum_m N_z(\mathbf{X}^{B^{(m)}})$, which is much smaller than $N_z(\mathbf{X})$ if there are repeating patterns in the feature matrix $\mathbf{X}$.

Now we can rewrite the predictive model in Equation 5.4 as follows

$$f(\{\mathcal{X}_n^{(v)}\}) = \sum_{v=1}^{V} \boldsymbol{\phi}^v \prod_{m \in S_M(v)} * \left(\mathbf{h}^{B^{(m)}}_{\psi(n)} + \mathbf{b}^{(v,m)\mathrm{T}} \right), \tag{5.11}$$

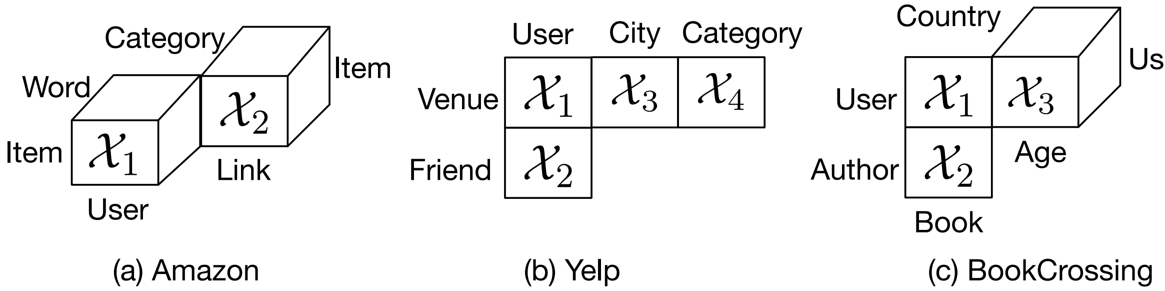
TABLE XVII

The statistics for each dataset. $N_z(X)$ and $N_z(\mathcal{B})$ are the number of non-zeros in plain formatted feature matrix and in relational structures, respectively. Game: Video Games, Cloth: Clothing, Shoes and Jewelry, Sport: Sports and Outdoors, Health: Health and Personal Care, Home: Home and Kitchen, Elec: Electronics.

Dataset	#Samples	Mode					Density	$N_z(X)$	$N_z(\mathcal{B})$
Amazon		#Users	#Items	#Words	#Categories	#Links			
Game	231,780	24,303	10,672	7,500	193	17,974	0.089%	32.9M	15.2M
Cloth	278,677	39,387	23,033	3,493	1,175	107,139	0.031%	25.6M	7.3M
Sport	296,337	35,598	18,357	5,202	1,432	73,040	0.045%	34.2M	10.2M
Health	346,355	38,609	18,534	5,889	849	80,379	0.048%	33.6M	12.1M
Home	551,682	66,569	28,237	6,455	970	99,090	0.029%	46.8M	19.4M
Elec	1,689,188	192,403	63,001	12,805	967	89,259	0.014%	161.5M	69M
		#Users	#Venues	#Friends	#Categories	#Cities			
Yelp	1,319,870	88,009	40,520	88,009	892	412	0.037%	70.5M	1.4M
		#Users	#Books	#Countries	#Ages	#Authors			
BX	244,848	24,325	45,074	57	8	17,178	0.022%	1.2M	163K

with the caches $\mathbf{H}^{B^{(m)}} = [\mathbf{h}_1^{B^{(m)}}, \cdots, \mathbf{h}_{N_m}^{B^{(m)}}]$ for each mode, where $\mathbf{h}_j^{B^{(m)}} = \mathbf{\Theta}^{(m)\mathrm{T}} \mathbf{x}_j^{B^{(m)}}$, $\forall j \in [1 : N_m]$.

This directly shows how N samples can be efficiently predicted: (i) compute $\mathbf{H}^{B^{(m)}}$ in $O(RN_z(\mathbf{X}^{B^{(m)}}))$ for each mode, (ii) compute N predictions with Equation 5.11 using caches in $O(RN(V + \sum_v M_v))$. With the help of relational structures, SFMs can learn and predict the same as in Equation 5.4 but with a much lower time complexity, especially for relational data with high cardinality.



(a) Amazon (b) Yelp (c) BookCrossing

Figure 15. Schema of the structural views in each dataset.

5.4 Experiments

5.4.1 Datasets

To evaluate the ability and applicability of the proposed SFMs, we include a spectrum of large datasets from different domains. The statistics for each dataset is summarized in Table XVII, the schema of the structural views in each dataset is presented in Figure 15, and the details are as follows:

Amazon[1]: The first group of datasets are from Amazon.com recently introduced by (McAuley et al., 2015). This is among the largest datasets available that include review texts and metadata of items. Each top category has been constructed as an independent dataset in (McAuley et al., 2015). In this Chapter, we take a variety of large categories as listed in Table XVII.

Each sample in these datasets has five modes, *i.e.*, users, items, review texts, categories, and linkage. The user mode and item mode are represented by one-hot encoding. The ℓ_2-

[1] http://jmcauley.ucsd.edu/data/amazon/

normalized TF-IDF vector representation of review text [1] of the item given by the user is used as the text mode. The category mode and linkage mode consists of all the categories and all the co-purchasing items of the item, which might be from other categories. The last two modes are ℓ_1-normalized.

Yelp[2]: It is a large-scale dataset consisting of venue reviews. Each sample in this dataset contains five modes, *i.e.*, users, venues, friends, categories and cities. The user mode and venue mode are represented by one-hot encoding. The friend mode consists of the friends' ids of users. The category mode and city mode consists of all the categories and the city of the venue. The last three modes are ℓ_1-normalized.

BookCrossing (BX)[3]: It is a book review dataset collected from the Book-Crossing community. Each sample in this dataset contains five modes, *i.e.*, users, books, countries, ages and authors. The ages are split in eight bins as in (Harper and Konstan, 2016). The country mode and age mode consist of the corresponding meta information of the user. The author modes represents the authors of the book. All the modes are represented by one-hot encoding.

The values of samples range within [1:5] in Amazon and Yelp datasets, and range within [1:10] in BX dataset.

[1] Stemming, lemmatization, removing stop-words and words with frequency less than 100 times, etc., are handled beforehand.

[2] https://www.yelp.com/dataset-challenge

[3] http://www2.informatik.uni-freiburg.de/∼cziegler/BX/

5.4.2 Comparison Methods

In order to demonstrate the effectiveness of the proposed SFMs, we compare a series of state-of-the-art methods.

Matrix Factorization (MF) is used to validate that meta information is helpful for improving prediction performance. We use the LIBMF implementation (Chin et al., 2016) for comparison in the experiment.

Factorization Machine (FM) (Rendle, 2012) is the state-of-the-art method in recommender systems. We compare with its higher-order extension (Blondel et al., 2016a) with up to second-order, and third-order feature interactions, and denote them as FM-2 and FM-3.

Polynomial Network (PolyNet) (Livni et al., 2014) is a recently proposed method that utilizes polynomial kernel on all features. We compare the augmented PolyNet (which adds a constant one to the feature vector (Blondel et al., 2016b)) with up to the second-order, and third-order kernel and denote them as PolyNet-2 and PolyNet-3.

Multi-View Machine (MVM) (Cao et al., 2016) is a tensor factorization based method that explores the latent representation embedded in the full-order interactions among all the modes.

Structural Factorization Machine (SFM) is the proposed model that learns the common latent spaces shared in multi-way relational data.

5.4.3 Experimental Settings

For each dataset, we randomly split 50%, 10%, and 40% of labeled samples as training set, validation set, and testing set, respectively. Validation sets are used for hyper-parameter tuning for each model. Each validation and testing set does not overlap with any other. For simplicity and fair comparison, in all the comparison methods, the dimension of latent factors $R = 20$ and the maximum number of epochs is set as 400 and we use early stop to obtain the best results for each method. Forbenius norm regularizers are used to avoid overfitting. The regularization hyper-parameter is tuned from $\{10^{-5},\ 10^{-4},\ \cdots,\ 10^{0}\}$.

All the methods except MF are implemented in TensorFlow, and the parameters are initialized using scaling variance initializer (He et al., 2015). We tune the scaling factor of initializer σ from $\{1, 2, 5, 10, 100\}$ and the learning rate η from $\{0.01, 0.1, 1\}$ using the validation sets. In the experiment, we set $\sigma = 2$ (default setting in TensorFlow) and $\eta = 0.01$ for these methods except MVM. We found that MVM is more sensitive to the configuration, because MVM will element-wisely multiply the latent factors of all the modes which leads to an extremely small value approaching zero. $\sigma = 10$ and $\eta = 0.1$ yielded the best performance for MVM.

To investigate the performance of comparison methods, we adopt mean squared error (MSE) on the test data as the evaluation metrics (McAuley and Leskovec, 2013; Zheng et al., 2017). The smaller value of the metric indicates the better performance. Each experiment was repeated for 10 times, and the mean and standard deviation of each metric in each data set were reported. All experiments are conducted on a single machine with Intel Xeon 6-Core CPUs of 2.4 GHz and equipped with a Maxwell Titan X GPU.

5.4.4 Performance Analysis

The experimental results are shown in Table XVIII. The best method of each dataset is in bold. For clarity, on the right of the tables we show the percentage improvement of the proposed SFM method over a variety of methods. From these results, we can observe that SFM

TABLE XVIII

MSE comparison on all the datasets. The best results are listed in bold.

Dataset	(a) **MF**	(b) **MVM**	(c) **FM-2**	(d) **FM-3**	(e) **PolyNet-2**	(f) **PolyNet-3**	(g) **SFM**	Improvement of SFM verus b	 min(c,d)	 min(e,f)
Game	1.569 ± 0.005	0.753 ± 0.007	0.764 ± 0.006	0.749 ± 0.007	0.749 ± 0.004	0.748 ± 0.006	**0.723 ± 0.006**	4.06%	3.52%	3.35%
Cloth	1.624 ± 0.009	0.725 ± 0.046	0.678 ± 0.004	0.679 ± 0.004	0.678 ± 0.007	0.680 ± 0.005	**0.659 ± 0.013**	9.03%	2.82%	2.84%
Sport	1.290 ± 0.004	0.646 ± 0.019	0.638 ± 0.003	0.632 ± 0.007	0.631 ± 0.005	0.632 ± 0.005	**0.614 ± 0.011**	5.00%	2.91%	2.79%
Health	1.568 ± 0.007	0.807 ± 0.012	0.779 ± 0.004	0.778 ± 0.004	0.779 ± 0.005	0.776 ± 0.005	**0.763 ± 0.019**	5.47%	2.02%	1.77%
Home	1.591 ± 0.004	0.729 ± 0.067	0.714 ± 0.002	0.714 ± 0.004	0.690 ± 0.003	0.692 ± 0.005	**0.678 ± 0.008**	6.93%	5.00%	1.72%
Elec	1.756 ± 0.002	0.792 ± 0.042	0.776 ± 0.006	0.749 ± 0.007	0.760 ± 0.004	0.757 ± 0.001	**0.747 ± 0.006**	5.69%	0.27%	1.33%
Yelp	1.713 ± 0.003	1.2575 ± 0.013	1.277 ± 0.002	1.277 ± 0.002	1.272 ± 0.002	1.272 ± 0.002	**1.256 ± 0.010**	0.09%	1.58%	1.19%
BX	4.094 ± 0.025	2.844 ± 0.024	2.766 ± 0.012	2.767 ± 0.014	2.654 ± 0.013	2.658 ± 0.013	**2.541 ± 0.025**	10.66%	8.16%	4.27%
Average on all datasets								5.87%	3.29%	2.41%

consistently outperforms all the comparison methods. We also make a few comparisons and summarize our findings as follows.

Compared with MF, SFM performs better with an average improvement of nearly 50%. MF usually performs well in practice (Ling et al., 2014; Rendle, 2012), while in datasets which are extremely sparse, as is shown in our case, MF cannot learn an accurate representation of users and items. Thus, the performance of MF is much worse than the other methods that utilize the meta information.

In both FM and PolyNet methods, the feature vectors from all the modes are concatenated as a single input feature vector. The major difference between these two methods is the choice of kernel applied (Blondel et al., 2016a). The polynomial kernel used in PolyNet considers all monomials (the products of features), i.e., all combinations of features *with* replacement. The ANOVA kernel used in FM considers only monomials composed of distinct features, i.e., feature combinations *without* replacement. Compared with the best results obtained from FM methods and from PolyNet methods, SFM leads to an average improvement of 3.3% and 2.4% in MSE, respectively.

The primary reason behind the results is how the latent factors of each feature are learned. For any factorization based method, the latent factors of a feature are essentially learned from its interactions with other features observed in the data, as can be observed from its update rule. In FM and PolyNet, all the feature interactions are taken into consideration without distinguishing the features from different modes. As a result, important feature interactions (e.g., the interactions between the given user and her friend) would be easily buried in irrelevant feature interactions from the same modes (e.g., the interactions between the friends of the same user). Hence, the learned latent factors are less representative in FM and PolyNet, compared with the proposed SFM. Besides, we can find that including higher-order interactions in FM and PolyNet (i.e., FM-3 and PolyNet-3) does not always improve the performance. Instead, it may even degrade the performance, as shown in Cloth, Yelp, and BX datasets. This is probably due to overfitting, as they need to include more parameters to model the interactions in higher orders while the datasets are extremely sparse such that the parameters cannot be properly learned.

Compared to the MVM method, which models the full-order interactions among all the modes, our proposed SFM leads to an average improvement of 5.87%. This is because not all the modes are relevant, and some irrelevant feature interactions may introduce noises to the learning task, which could be further exaggerated after combinations of interactions. This suggests that preserving the nature of relational structure is important in building predictive models.

5.4.5 Computational Cost Analysis

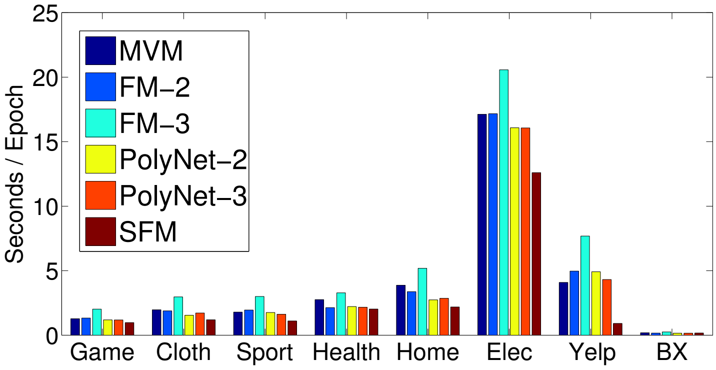


Figure 16. Training Time (Seconds/Epoch) Comparison.

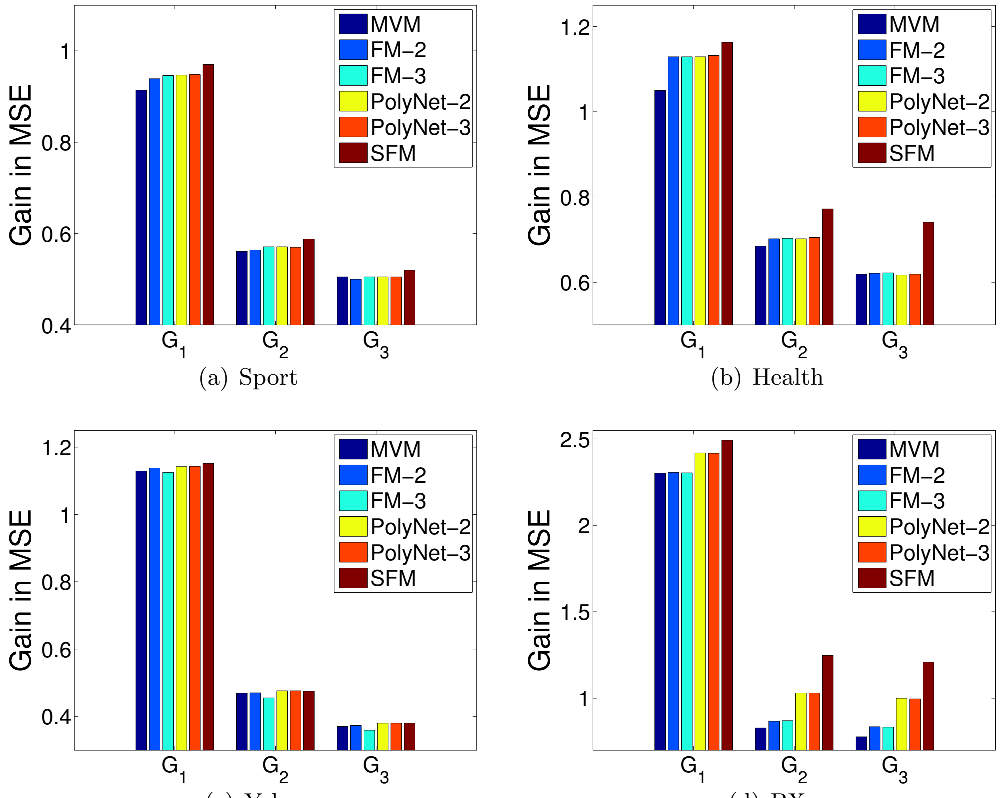
Next, we investigate the computational cost for comparison methods. The averaged training time (seconds per epoch) required for each dataset is shown in Figure 16. We can easily find that the proposed SFM requires much less computational cost on all the datasets, especially for the Yelp dataset (roughly 11% of computational cost required for training FM-3). The efficiency comes from the use of relational structure representation. As shown in Table XVII, the number of non-zeros of the feature matrix $N_z(\mathbf{X})$ is much larger than the number of non-zeros of the relational structure representation $N_z(\mathcal{B})$. The amount of repeating patterns is much higher for the Yelp dataset than for the other dataset, because adding all the friends of a user significantly increases results in large repeating blocks in the plain feature matrix. Standard ML algorithms like the compared methods have typically at best a linear complexity in $N_z(\mathbf{X})$, while using the relational structure representation for SFM have a linear complexity in $N_z(\mathcal{B})$. This experiment substantiates the efficiency of the proposed SFM for large datasets.

5.4.6 Analysis of the Impact of Data Sparsity

We proceed by further studying the impact of data sparsity on different methods. For datasets that are sparse, it can be easily found that the improvement of SFM over MF is significant, mainly because the number of samples is too scarce to model the items and users adequately. In order to verify this finding, we compare the performance of MF with all the other methods on the set of users with limited training samples. The gain of each method over MF is shown in Figure 17, where G_1, G_2, and G_3 are groups of users with $[1, 3]$, $[4, 6]$, and $[7, 10]$ observed samples in the training set. Due to space limit, we only report the results from two Amazon datasets (Sport and Health) while the observations still hold for the rest datasets. It can be seen that the proposed SFM gains the most in group G_1, in which the users have extremely few training items, and the performance gain decreases along with the number of training samples. The results indicate that including meta information can be valuable information especially when limited information available.

5.4.7 Sensitivity analysis

The number of latent factors R is an important hyperparameter for the factorization models. We analyze different values of R and report the averaged results in Figure 18. The results again

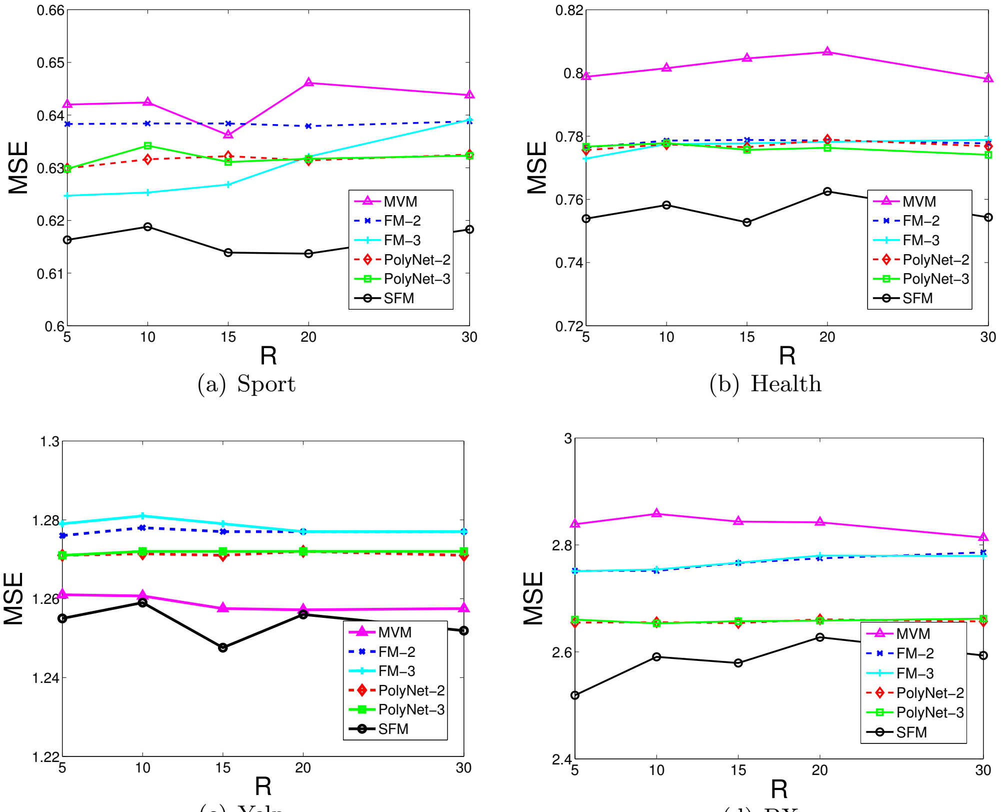


(a) Sport (b) Health

(c) Yelp (d) BX

Figure 17. Performance gain in MSE compared with MF for users with limited training samples. G_1, G_2, and G_3 are groups of users with $[1, 3]$, $[4, 6]$, and $[7, 10]$ observed samples in the training set, respectively.

show that SFM consistently outperforms other methods with various values of R. In contrast to findings in other related factorization models (Yan et al., 2014) where prediction error can steadily get reduced with larger R, we observe that the performance of each method is rather stable even with the increasing of R. It is reasonable in a general sense, as the expressiveness of the model is enough to describe the information embedded in data. Although larger R



(a) Sport

(b) Health

(c) Yelp

(d) BX

Figure 18. Sensitivity analysis of the latent dimension R.

renders the model with greater expressiveness, when the available observations regarding the target values are too sparse but the meta information is rich, only a few number of factors are required to fit the data well.

5.5 Related Work

Rendle pioneered the concept of feature interactions in Factorization Machines (FM) (Rendle, 2012). Juan et al. presented Field-aware Factorization Machines (FFM) (Juan et al., 2016) to allow each feature to interact differently with another feature depending on its field. Novikov et al. proposed Exponential Machines (ExM) (Novikov et al., 2017) where the weight tensor is represented in a factorized format called Tensor Train. Zhang et al. used FM to initialize the embedding layer in a deep model (Zhang et al., 2016). Qu et al. added a product layer on the top of the embedding layer to increase the model capacity (Qu et al., 2016). Other extensions of FM to deep architectures include Neural Factorization Machines (NFM) (He and Chua, 2017) and Attentional Factorization Machines (AFM) (Xiao et al., 2017). In order to effectively model feature interactions, a variety of models has been developed in the industry as well. Microsoft studied feature interactions in deep models, including Deep Semantic Similarity Model (DSSM) (Huang et al., 2013), Deep Crossing (Shan et al., 2016) and Deep Embedding Forest (Zhu et al., 2017). They use features as raw as possible without manually crafted combinatorial features, and let deep neural networks take care of the rest. Alibaba proposed a Deep Interest Network (DIN) (Zhou et al., 2017) to learn user embeddings as a function of ad embeddings. Google used deep neural networks to learn from heterogeneous signals for YouTube recommendations (Covington et al., 2016). In addition, Wide & Deep Models (Cheng et al., 2016) were developed for app recommender systems in Google Play where the wide component includes cross features that are good at memorization and the deep component includes embedding layers for generalization. Guo et al. proposed to use FM as the wide component in Wide & Deep with shared embeddings in the deep component (Guo et al., 2017). Wang et al. developed the Deep & Cross Network (DCN) to learn explicit cross features of bounded degree (Wang et al., 2017). However, previous approaches will introduce unexpected noise from the irrelevant feature interactions that can even be exaggerated after combinations, thereby degrading performance as demonstrated in the experiments. Different from conventional approaches, the proposed algorithm can learn the common latent spaces shared in multi-view tensors and automatically adjusting the importance of each view in the predictive model.

CHAPTER 6

CONCLUSION

(Part of the chapter was previously published in (Lu et al., 2014a; Lu et al., 2016; Lu et al., 2017; Lu et al., 2018).)

In this dissertation, we have discussed broad learning in multiple heterogeneous domains. We covered two major approaches in broad learning: network-based approaches and matrix/tensor factorization based approaches. Towards this direction, we thoroughly studied four different research problems: connecting heterogeneous networks, transfer learning for new domains, multi-task multi-view learning, and modeling multi-view relational data. We evaluated the effectiveness of the proposed models and algorithms by extensive experiments on various real-world datasets. The major contributions are summarized as follows.

First, we have described and studied the problem of connecting heterogeneous networks. Different from previous works in link prediction and network alignment, it requires to predict links between accounts across partially aligned networks with completely different schema. We have proposed to extract two types of features, user profile features and user interest features, that can be used to compute the similarity scores of pairs across such networks. By finding the top-K maximum similar and stable matching, our proposed approach can effectively connect user accounts across heterogeneous networks. Extensively experiments have demonstrated that the proposed method consistently outperforms other commonly-used baselines. It provides a promising step towards incorporating existing online social networks for e-commence.

Additionally, we explored how to leverage knowledge from related domains to fit the newly-emerged domain. We have proposed a novel similarity measure called *AmpSim* that can judiciously capture the rich similarity semantics between entities by taking both the linkage structures and the augmented link attributes into account. We further incorporated the similarity information captured by AmpSim in a constrained collective matrix factorization model. Extensively experiments on real-world datasets have demonstrated that our proposed model significantly outperforms other state-of-the-art collaborative filtering algorithms in addressing item recommendation for emerging domain.

Furthermore, we also studied a relatively new research direction in multi-task multi-view (MTMV) learning. We have presented efficient multilinear factorization machines (MFMs) that can learn the task-specific features and the common latent spaces embedded within the multimodal interactions among the multiple tasks and the multiple views. Because full-order interactions are collectively used during learning procedure, MFMs can deal with the partially incomplete data without difficulty. Moreover, the complexity of MFMs is linear in the number of features, which make MFMs suitable to large-scale real-world problems. Extensive experiments on four real-world datasets demonstrated that our proposed MFMs outperform several state-of-the-art methods in a wide variety of MTMV learning problems.

Finally, we introduced a generic framework for learning relational data from heterogeneous domains, which can explore the high order correlations underlying multi-view relational data. We developed structural factorization machines (SFMs) that learn the common latent spaces shared in the multi-view tensors while automatically adjust the contribution of each view in the predictive model. With the help of relational structure representation, we further provided an efficient approach to avoid unnecessary computation costs on repeating patterns of the multi-view data. It was shown that the proposed SFMs outperform state-of-the-art factorization models on eight large-scale datasets in terms of prediction accuracy and computational cost.

APPENDICES

.1 ACM Copyright Letter

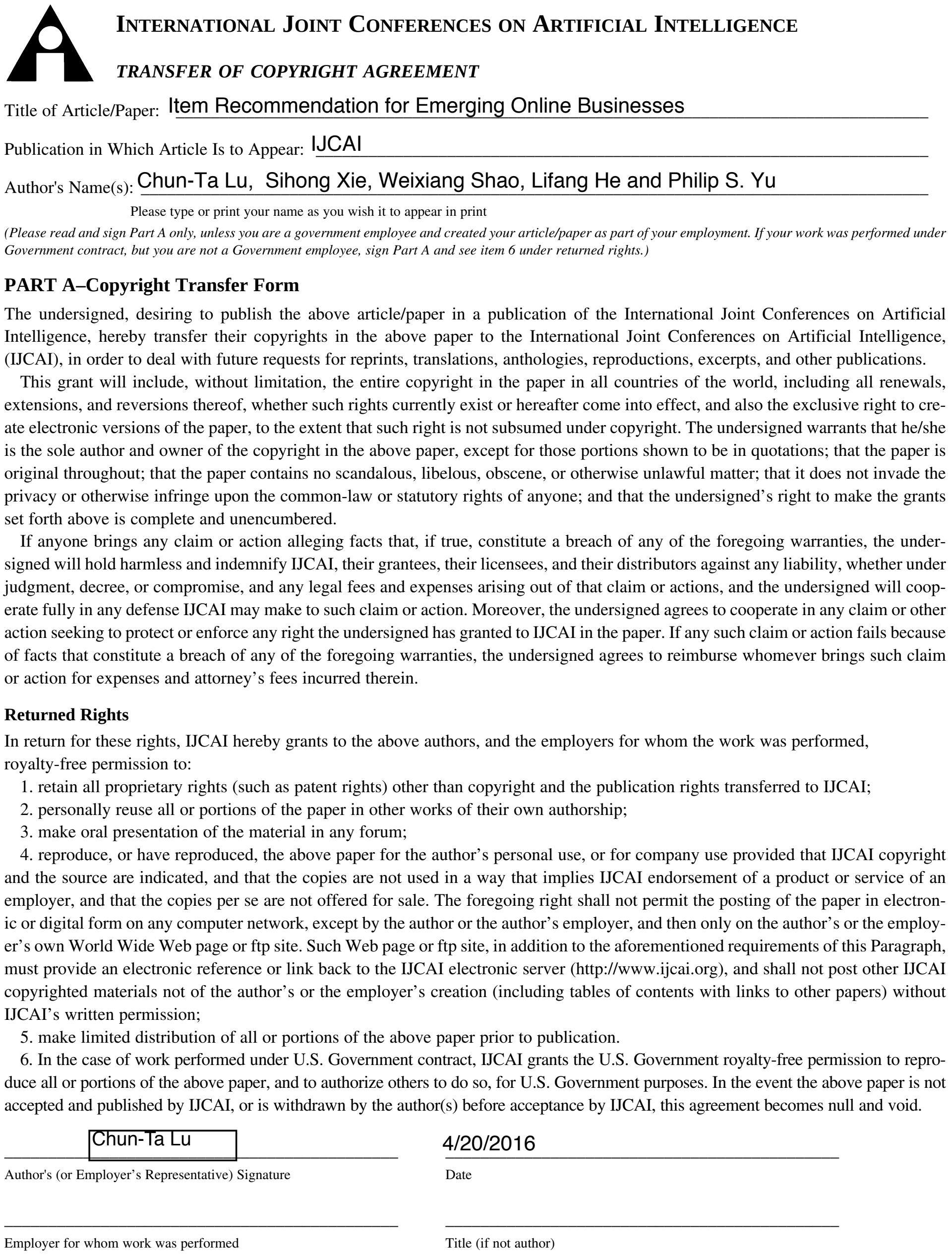
“Authors can reuse any portion of their own work in a new work of their own (and no fee is expected) as long as a citation and DOI pointer to the Version of Record in the ACM Digital Library are included.

Contributing complete papers to any edited collection of reprints for which the author is not the editor, requires permission and usually a republication fee.

Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included. Authors can use any portion of their own work in presentations and in the classroom (and no fee is expected).” [1]

[1] http://authors.acm.org/main.html

.2 IJCAI Copyright Letter



INTERNATIONAL JOINT CONFERENCES ON ARTIFICIAL INTELLIGENCE

TRANSFER OF COPYRIGHT AGREEMENT

Title of Article/Paper: Item Recommendation for Emerging Online Businesses

Publication in Which Article Is to Appear: IJCAI

Author's Name(s): Chun-Ta Lu, Sihong Xie, Weixiang Shao, Lifang He and Philip S. Yu

Please type or print your name as you wish it to appear in print

(Please read and sign Part A only, unless you are a government employee and created your article/paper as part of your employment. If your work was performed under Government contract, but you are not a Government employee, sign Part A and see item 6 under returned rights.)

PART A–Copyright Transfer Form

The undersigned, desiring to publish the above article/paper in a publication of the International Joint Conferences on Artificial Intelligence, hereby transfer their copyrights in the above paper to the International Joint Conferences on Artificial Intelligence, (IJCAI), in order to deal with future requests for reprints, translations, anthologies, reproductions, excerpts, and other publications.

This grant will include, without limitation, the entire copyright in the paper in all countries of the world, including all renewals, extensions, and reversions thereof, whether such rights currently exist or hereafter come into effect, and also the exclusive right to create electronic versions of the paper, to the extent that such right is not subsumed under copyright. The undersigned warrants that he/she is the sole author and owner of the copyright in the above paper, except for those portions shown to be in quotations; that the paper is original throughout; that the paper contains no scandalous, libelous, obscene, or otherwise unlawful matter; that it does not invade the privacy or otherwise infringe upon the common-law or statutory rights of anyone; and that the undersigned's right to make the grants set forth above is complete and unencumbered.

If anyone brings any claim or action alleging facts that, if true, constitute a breach of any of the foregoing warranties, the undersigned will hold harmless and indemnify IJCAI, their grantees, their licensees, and their distributors against any liability, whether under judgment, decree, or compromise, and any legal fees and expenses arising out of that claim or actions, and the undersigned will cooperate fully in any defense IJCAI may make to such claim or action. Moreover, the undersigned agrees to cooperate in any claim or other action seeking to protect or enforce any right the undersigned has granted to IJCAI in the paper. If any such claim or action fails because of facts that constitute a breach of any of the foregoing warranties, the undersigned agrees to reimburse whomever brings such claim or action for expenses and attorney's fees incurred therein.

Returned Rights

In return for these rights, IJCAI hereby grants to the above authors, and the employers for whom the work was performed, royalty-free permission to:

1. retain all proprietary rights (such as patent rights) other than copyright and the publication rights transferred to IJCAI;
2. personally reuse all or portions of the paper in other works of their own authorship;
3. make oral presentation of the material in any forum;
4. reproduce, or have reproduced, the above paper for the author's personal use, or for company use provided that IJCAI copyright and the source are indicated, and that the copies are not used in a way that implies IJCAI endorsement of a product or service of an employer, and that the copies per se are not offered for sale. The foregoing right shall not permit the posting of the paper in electronic or digital form on any computer network, except by the author or the author's employer, and then only on the author's or the employer's own World Wide Web page or ftp site. Such Web page or ftp site, in addition to the aforementioned requirements of this Paragraph, must provide an electronic reference or link back to the IJCAI electronic server (http://www.ijcai.org), and shall not post other IJCAI copyrighted materials not of the author's or the employer's creation (including tables of contents with links to other papers) without IJCAI's written permission;
5. make limited distribution of all or portions of the above paper prior to publication.
6. In the case of work performed under U.S. Government contract, IJCAI grants the U.S. Government royalty-free permission to reproduce all or portions of the above paper, and to authorize others to do so, for U.S. Government purposes. In the event the above paper is not accepted and published by IJCAI, or is withdrawn by the author(s) before acceptance by IJCAI, this agreement becomes null and void.

Chun-Ta Lu	4/20/2016
Author's (or Employer's Representative) Signature	Date
Employer for whom work was performed	Title (if not author)

CITED LITERATURE

[Acar et al. , 2011]Acar, E., Kolda, T. G., and Dunlavy, D. M.: All-at-once optimization for coupled matrix and tensor factorizations. arXiv preprint arXiv:1105.3422, 2011.

[Adamic and Adar, 2003]Adamic, L. A. and Adar, E.: Friends and neighbors on the web. Social Networks, 25(3):211–230, 2003.

[Ando and Zhang, 2005]Ando, R. K. and Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. JMLR, 6:1817–1853, 2005.

[Antonellis et al. , 2008]Antonellis, I., Molina, H. G., and Chang, C. C.: Simrank++: Query rewriting through link analysis of the click graph. Proc. VLDB Endow., 1(1):408–421, 2008.

[Argyriou et al. , 2008]Argyriou, A., Evgeniou, T., and Pontil, M.: Convex multi-task feature learning. Machine Learning, 73(3):243–272, 2008.

[Aylward et al. , 1995]Aylward, E. H., Brettschneider, P. D., McArthur, J. C., Harris, G. J., Schlaepfer, T. E., Henderer, J. D., Barta, P. E., Tien, A. Y., and Pearlson, G. D.: Magnetic resonance imaging measurement of gray matter volume reductions in HIV dementia. The American journal of psychiatry, 152(7):987–994, 1995.

[Backstrom and Leskovec, 2011]Backstrom, L. and Leskovec, J.: Supervised random walks: predicting and recommending links in social networks. In WSDM, pages 635–644, 2011.

[Bayati et al. , 2009]Bayati, M., Gerritsen, M., Gleich, D., Saberi, A., and Wang, Y.: Algorithms for large, sparse network alignment problems. In ICDM, pages 705–710, 2009.

[Bayati et al. , 2013]Bayati, M., Gleich, D. F., Saberi, A., and Wang, Y.: Message-passing algorithms for sparse network alignment. TKDD, 7(1):3, 2013.

[Belkin and Niyogi, 2001]Belkin, M. and Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In NIPS, pages 585–591, 2001.

[Benchettara et al. , 2010]Benchettara, N., Kanawati, R., and Rouveirol, C.: Supervised machine learning applied to link prediction in bipartite social networks. In ASONAM, pages 326–330, 2010.

[Berlinet and Thomas-Agnan, 2011]Berlinet, A. and Thomas-Agnan, C.: Reproducing kernel Hilbert spaces in probability and statistics. Springer Science & Business Media, 2011.

[Bhagat et al. , 2012]Bhagat, S., Goyal, A., and Lakshmanan, L. V. S.: Maximizing product adoption in social networks. In WSDM, pages 603–612, 2012.

[Bhatt et al. , 2010]Bhatt, R., Chaoji, V., and Parekh, R.: Predicting product adoption in large-scale social networks. In CIKM, pages 1039–1048, 2010.

[Bhattacharya and Getoor, 2007]Bhattacharya, I. and Getoor, L.: Collective entity resolution in relational data. TKDD, 1(1), 2007.

[Blondel et al. , 2016a]Blondel, M., Fujino, A., Ueda, N., and Ishihata, M.: Higher-order factorization machines. In NIPS, pages 3351–3359, 2016.

[Blondel et al. , 2016b]Blondel, M., Ishihata, M., Fujino, A., and Ueda, N.: Polynomial networks and factorization machines: New insights and efficient training algorithms. In ICML, pages 850–858, 2016.

[Blum and Mitchell, 1998]Blum, A. and Mitchell, T.: Combining labeled and unlabeled data with co-training. In COLT, pages 92–100, 1998.

[Brenchley et al. , 2004]Brenchley, J. M., Schacker, T. W., Ruff, L. E., Price, D. A., Taylor, J. H., Beilman, G. J., Nguyen, P. L., Khoruts, A., Larson, M., Haase, A. T., et al.: CD4+ T cell depletion during all stages of HIV disease occurs predominantly in the gastrointestinal tract. The Journal of experimental medicine, 200(6):749–759, 2004.

[Brown and Reingen, 1987]Brown, J. J. and Reingen, P. H.: Social ties and word-of-mouth referral behavior. Journal of Consumer Research, 14(3):350–62, December 1987.

[Cai et al. , 2011]Cai, D., He, X., Han, J., and Huang, T. S.: Graph regularized nonnegative matrix factorization for data representation. TPAMI, 33(8):1548–1560, 2011.

[Cao et al. , 2014a]Cao, B., He, L., Kong, X., Yu, P. S., Hao, Z., and Ragin, A. B.: Tensor-based multi-view feature selection with applications to brain diseases. In ICDM, pages 40–49, 2014.

[Cao et al. , 2014b]Cao, B., Kong, X., and Yu, P. S.: Collective prediction of multiple types of links in heterogeneous information networks. In *ICDM*, pages 50–59, 2014.

[Cao et al. , 2017]Cao, B., Zheng, L., Zhang, C., Yu, P. S., Piscitello, A., Zulueta, J., Ajilore, O., Ryan, K., and Leow, A. D.: Deepmood: Modeling mobile phone typing dynamics for mood detection. In *SIGKDD*, pages 747–755. ACM, 2017.

[Cao et al. , 2016]Cao, B., Zhou, H., Li, G., and Yu, P. S.: Multi-view machines. In *WSDM*, 2016.

[Caruana, 1998]Caruana, R.: Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[Ceci et al. , 2012]Ceci, M., Appice, A., Viktor, H. L., Malerba, D., Paquet, E., and Guo, H.: Transductive relational classification in the co-training paradigm. In *Workshop on MLDM*, pages 11–25. Springer, 2012.

[Chang and Lin, 2011]Chang, C.-C. and Lin, C.-J.: LIBSVM: A library for support vector machines. *ACM TIST*, 2:27:1–27:27, 2011.

[Chen et al. , 2009]Chen, J., Tang, L., Liu, J., and Ye, J.: A convex formulation for learning shared structures from multiple tasks. In *ICML*, pages 137–144, 2009.

[Chen et al. , 2011]Chen, J., Zhou, J., and Ye, J.: Integrating low-rank and group-sparse structures for robust multi-task learning. In *SIGKDD*, pages 42–50, 2011.

[Chen et al. , 2010]Chen, N., Zhu, J., and Xing, E. P.: Predictive subspace learning for multiview data: a large margin approach. In *NIPS*, pages 361–369, 2010.

[Cheng et al. , 2016]Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al.: Wide & deep learning for recommender systems. In *DLRS*, pages 7–10. ACM, 2016.

[Chin et al. , 2016]Chin, W.-S., Yuan, B.-W., Yang, M.-Y., Zhuang, Y., Juan, Y.-C., and Lin, C.-J.: Libmf: A library for parallel matrix factorization in shared-memory systems. *JMLR*, 17(1):2971–2975, 2016.

[Chong and Zak, 2013]Chong, E. K. and Zak, S. H.: *An introduction to optimization*, volume 76. John Wiley & Sons, 2013.

[Chowdhury et al. , 1990]Chowdhury, I. H., Munakata, T., Koyanagi, Y., Kobayashi, S., Arai, S., and Yamamoto, N.: Mycoplasma can enhance HIV replication in vitro: a possible cofactor responsible for the progression of AIDS. Biochemical and biophysical research communications, 170(3):1365–1370, 1990.

[Christoudias et al. , 2008]Christoudias, C. M., Urtasun, R., and Darrell, T.: Unsupervised feature selection via distributed coding for multi-view object recognition. In CVPR, pages 1–8, 2008.

[Chua et al. , 2013]Chua, F. C. T., Lauw, H. W., and Lim, E.-P.: Generative models for item adoptions using social correlation. TKDE, 25(9):2036–2048, 2013.

[Chua et al. , 2009]Chua, T.-S., Tang, J., Hong, R., Li, H., Luo, Z., and Zheng, Y.-T.: Nus-wide: A real-world web image database from national university of singapore. In CIVR, 2009.

[Cocchi et al. , 1995]Cocchi, F., DeVico, A. L., Garzino-Demo, A., Arya, S. K., Gallo, R. C., and Lusso, P.: Identification of RANTES, MIP-1α, and MIP-1β as the major HIV-suppressive factors produced by CD8+ T cells. Science, 270(5243):1811–1815, 1995.

[Cortes et al. , 2009]Cortes, C., Mohri, M., and Rostamizadeh, A.: Learning non-linear combinations of kernels. In NIPS, pages 396–404, 2009.

[Cortes and Vapnik, 1995]Cortes, C. and Vapnik, V.: Support-vector networks. Machine learning, 20(3):273–297, 1995.

[Covington et al. , 2016]Covington, P., Adams, J., and Sargin, E.: Deep neural networks for youtube recommendations. In RecSys, pages 191–198. ACM, 2016.

[Crandall et al. , 2008]Crandall, D. J., Cosley, D., Huttenlocher, D. P., Kleinberg, J. M., and Suri, S.: Feedback effects between similarity and social influence in online communities. In SIGKDD, pages 160–168, 2008.

[Cristianini and Shawe-Taylor, 2000]Cristianini, N. and Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods. Cambridge university press, 2000.

[Dash and Liu, 1997]Dash, M. and Liu, H.: Feature selection for classification. Intelligent data analysis, 1(3):131–156, 1997.

[De Lathauwer et al. , 2000]De Lathauwer, L., De Moor, B., and Vandewalle, J.: On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors. SIAM Journal on Matrix Analysis and Applications, 21(4):1324–1342, 2000.

[Dice, 1945]Dice, L. R.: Measures of the Amount of Ecologic Association Between Species. Ecology, 26(3):297–302, 1945.

[DiMaggio and Louch, 1998]DiMaggio, P. J. and Louch, H.: Socially Embedded Consumer Transactions: For What Kinds of Purchases Do People Most Often use Networks? American Sociological Review, 63(5):619–637, October 1998.

[Ding et al. , 2006]Ding, C. H. Q., Li, T., and Peng, W.: Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method. In AAAI, pages 342–347, 2006.

[Dubins and Freedman, 1981]Dubins, L. E. and Freedman, D. A.: Machiavelli and the Gale-Shapley algorithm. American Mathematical Monthly, 88(7):485–494, 1981.

[Duchi et al. , 2011]Duchi, J., Hazan, E., and Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. JMLR, 12:2121–2159, 2011.

[Dvir et al. , 2003]Dvir, D., Raz, T., and Shenhar, A. J.: An empirical analysis of the relationship between project planning and project success. International Journal of Project Management, 21(2):89–95, 2003.

[Evgeniou and Pontil, 2007]Evgeniou, A. and Pontil, M.: Multi-task feature learning. NIPS, 19:41, 2007.

[Evgeniou et al. , 2005]Evgeniou, T., Micchelli, C. A., and Pontil, M.: Learning multiple tasks with kernel methods. In JMLR, pages 615–637, 2005.

[Evgeniou and Pontil, 2004]Evgeniou, T. and Pontil, M.: Regularized multi–task learning. In SIGKDD, pages 109–117, 2004.

[Faddoul et al. , 2010]Faddoul, J. B., Chidlovskii, B., Torre, F., and Gilleron, R.: Boosting multi-task weak learners with applications to textual and social data. In ICMLA, pages 367–372. IEEE, 2010.

[Fan et al. , 2008]Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J.: Liblinear: A library for large linear classification. JMLR, 9:1871–1874, 2008.

[Fang and Zhang, 2013]Fang, Z. and Zhang, Z. M.: Discriminative feature selection for multi-view cross-domain learning. In CIKM, pages 1321–1330, 2013.

[Feng et al. , 2012]Feng, Y., Xiao, J., Zhuang, Y., and Liu, X.: Adaptive unsupervised multi-view feature selection for visual concept recognition. In ACCV, pages 343–357, 2012.

[Fischer, 1982]Fischer, C. S.: To dwell among friends: Personal networks in town and city. Chicago, IL, University of Chicago Press, 1982.

[Friedman et al. , 2010]Friedman, J., Hastie, T., and Tibshirani, R.: A note on the group lasso and a sparse group lasso. arXiv, 2010.

[Gärtner, 2003]Gärtner, T.: A survey of kernels for structured data. SIGKDD Explorations, 5(1):49–58, 2003.

[Getoor and Diehl, 2005]Getoor, L. and Diehl, C. P.: Link mining: a survey. SIGKDD Explorations, 7(2):3–12, 2005.

[Gönen and Alpaydın, 2011]Gönen, M. and Alpaydın, E.: Multiple kernel learning algorithms. JMLR, 12:2211–2268, 2011.

[Gong et al. , 2012]Gong, P., Ye, J., and Zhang, C.: Robust multi-task feature learning. In SIGKDD, pages 895–903, 2012.

[Gonzalez et al. , 2014]Gonzalez, J. E., Xin, R. S., Dave, A., Crankshaw, D., Franklin, M. J., and Stoica, I.: GraphX: Graph processing in a distributed dataflow framework. In OSDI, pages 599–613. USENIX, 2014.

[Guo and Viktor, 2006]Guo, H. and Viktor, H. L.: Mining relational data through correlation-based multiple view validation. In SIGKDD, pages 567–573. ACM, 2006.

[Guo et al. , 2017]Guo, H., Tang, R., Ye, Y., Li, Z., and He, X.: Deepfm: A factorization-machine based neural network for ctr prediction. arXiv preprint arXiv:1703.04247, 2017.

[Guo et al. , 2011]Guo, S., Wang, M., and Leskovec, J.: The role of social networks in online shopping: information passing, price of trust, and consumer choice. In ACM EC, pages 157–166, 2011.

[Guyon et al. , 2002]Guyon, I., Weston, J., Barnhill, S., and Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine learning, 46(1-3):389–422, 2002.

[Harper and Konstan, 2015]Harper, F. M. and Konstan, J. A.: The movielens datasets: History and context. TiiS, 5(4):19, 2015.

[Harper and Konstan, 2016]Harper, F. M. and Konstan, J. A.: The movielens datasets: History and context. TiiS, 5(4):19, 2016.

[He and Lawrence, 2011]He, J. and Lawrence, R.: A graph-based framework for multi-task multi-view learning. In ICML, pages 25–32, 2011.

[He et al. , 2015]He, K., Zhang, X., Ren, S., and Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In ICCV, pages 1026–1034, 2015.

[He et al. , 2014]He, L., Kong, X., Yu, P. S., Ragin, A. B., Hao, Z., and Yang, X.: DuSK: A dual structure-preserving kernel for supervised tensor learning with applications to neuroimages. In SDM. SIAM, 2014.

[He et al. , 2016]He, L., Lu, C.-T., Ma, J., Cao, J., Shen, L., and Yu, P. S.: Joint community and structural hole spanner detection via harmonic modularity. In SIGKDD, pages 875–884. ACM, 2016.

[He and Chua, 2017]He, X. and Chua, T.-S.: Neural factorization machines for sparse predictive analytics. In SIGIR, 2017.

[Hill et al. , 2006]Hill, S., Provost, F., and Volinsky, C.: Network-based marketing: Identifying likely adopters via consumer networks. Statistical Science, 22(2):256–275, 2006.

[Hinton et al. , 2012]Hinton, G., Srivastava, N., and Swersky, K.: Rmsprop: Divide the gradient by a running average of its recent magnitude. Neural networks for machine learning, Coursera lecture 6e, 2012.

[Hong et al. , 2013]Hong, L., Doumith, A. S., and Davison, B. D.: Co-factorization machines: modeling user interests and predicting individual decisions in twitter. In WSDM, pages 557–566, 2013.

[Hu and Liu, 2004]Hu, M. and Liu, B.: Mining and summarizing customer reviews. In SIGKDD, pages 168–177, 2004.

[Huang et al. , 2013]Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., and Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In CIKM, pages 2333–2338. ACM, 2013.

[Jacob et al. , 2009]Jacob, L., Vert, J.-P., and Bach, F. R.: Clustered multi-task learning: A convex formulation. In NIPS, pages 745–752, 2009.

[Jamali and Ester, 2009]Jamali, M. and Ester, M.: Trustwalker: a random walk model for combining trust-based and item-based recommendation. In SIGKDD, pages 397–406, 2009.

[Jamali and Ester, 2010]Jamali, M. and Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In RecSys, pages 135–142, 2010.

[Jamali et al. , 2011]Jamali, M., Huang, T., and Ester, M.: A generalized stochastic block model for recommendation in social rating networks. In RecSys, pages 53–60, Chicago, IL, USA, 2011.

[Jeh and Widom, 2002]Jeh, G. and Widom, J.: Simrank: a measure of structural-context similarity. In SIGKDD, pages 538–543, 2002.

[Jeh and Widom, 2003]Jeh, G. and Widom, J.: Scaling personalized web search. In WWW, pages 271–279, 2003.

[Jiang et al. , 2012a]Jiang, M., Cui, P., Liu, R., Yang, Q., Wang, F., Zhu, W., and Yang, S.: Social contextual recommendation. In CIKM, pages 45–54, 2012.

[Jiang et al. , 2012b]Jiang, M., Cui, P., Wang, F., Yang, Q., Zhu, W., and Yang, S.: Social recommendation across multiple relational domains. In CIKM, pages 1422–1431, 2012.

[Jin et al. , 2013]Jin, X., Zhuang, F., Wang, S., He, Q., and Shi, Z.: Shared structure learning for multiple tasks with multiple views. In ECML/PKDD, pages 353–368, 2013.

[Jin et al. , 2014]Jin, X., Zhuang, F., Xiong, H., Du, C., Luo, P., and He, Q.: Multi-task multi-view learning for heterogeneous tasks. In CIKM, pages 441–450. ACM, 2014.

[John et al. , 1994]John, G. H., Kohavi, R., and Pfleger, K.: Irrelevant features and the subset selection problem. In ICML, pages 121–129, 1994.

[Juan et al. , 2015]Juan, Y.-C., Zhuang, Y., and Chin, W.-S.: LIBFFM: A Library for Field-aware Factorization Machines, 2015. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libffm`.

[Juan et al. , 2016]Juan, Y., Zhuang, Y., Chin, W.-S., and Lin, C.-J.: Field-aware factorization machines for ctr prediction. In RecSys, pages 43–50. ACM, 2016.

[Kabbur et al. , 2013]Kabbur, S., Ning, X., and Karypis, G.: Fism: factored item similarity models for top-n recommender systems. In SIGKDD, pages 659–667. ACM, 2013.

[Katz and Katz, 1953]Katz, L. and Katz, L.: A new status index derived from sociometric analysis. Psychometrika, 18(1):39–43, March 1953.

[Kim and Xing, 2010]Kim, S. and Xing, E. P.: Tree-guided group lasso for multi-task regression with structured sparsity. In ICML, pages 543–550, USA, 2010. Omnipress.

[Kingma and Ba, 2014]Kingma, D. and Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[Kolda and Bader, 2009]Kolda, T. G. and Bader, B. W.: Tensor decompositions and applications. SIAM review, 51(3):455–500, 2009.

[Kolda, 2006]Kolda, T. G.: Multilinear operators for higher-order decompositions. United States. Department of Energy, 2006.

[Kong et al. , 2013]Kong, X., Cao, B., and Yu, P. S.: Multi-label classification by mining label and instance correlations from heterogeneous information networks. In SIGKDD, pages 614–622, 2013.

[Kong et al. , 2011]Kong, X., Shi, X., and Philip, S. Y.: Multi-label collective classification. In SDM, volume 11, pages 618–629. SIAM, 2011.

[Kong and Yu, 2014]Kong, X. and Yu, P. S.: Brain network analysis: a data mining perspective. SIGKDD Explorations, 15(2):30–38, 2014.

[Kong et al. , 2013a]Kong, X., Yu, P. S., Wang, X., and Ragin, A. B.: Discriminative feature selection for uncertain graph classification. In SDM, 2013.

[Kong et al. , 2013b]Kong, X., Zhang, J., and Yu, P. S.: Inferring anchor links across multiple heterogeneous social networks. In CIKM, pages 179–188, 2013.

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

[Koren, 2010]Koren, Y.: Factor in the neighbors: Scalable and accurate collaborative filtering. TKDD, 4(1):1, 2010.

[Koutra et al. , 2013]Koutra, D., Tong, H., and Lubensky, D.: Big-align: Fast bipartite graph alignment. In ICDM, pages 389–398, 2013.

[Kumar et al. , 2011]Kumar, A., Rai, P., and Daume, H.: Co-regularized multi-view spectral clustering. In NIPS, eds. J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, pages 1413–1421. Curran Associates, Inc., 2011.

[Lanckriet et al. , 2004a]Lanckriet, G. R., Cristianini, N., Bartlett, P., Ghaoui, L. E., and Jordan, M. I.: Learning the kernel matrix with semidefinite programming. JMLR, 5:27–72, 2004.

[Lanckriet et al. , 2004b]Lanckriet, G. R., De Bie, T., Cristianini, N., Jordan, M. I., and Noble, W. S.: A statistical framework for genomic data fusion. Bioinformatics, 20(16):2626–2635, 2004.

[Lanckriet et al. , 2004c]Lanckriet, G. R., Deng, M., Cristianini, N., Jordan, M. I., and Noble, W. S.: Kernel-based data fusion and its application to protein function prediction in yeast. In Pacific symposium on biocomputing, pages 300–311, 2004.

[Lee and Seung, 2001]Lee, D. D. and Seung, H. S.: Algorithms for non-negative matrix factorization. In NIPS, pages 556–562, 2001.

[Leskovec et al. , 2006]Leskovec, J., Adamic, L. A., and Huberman, B. A.: The dynamics of viral marketing. In ACM EC, pages 228–237, 2006.

[Levenshtein, 1966]Levenshtein, V. I.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady., 10(8):707–710, February 1966.

[Li et al. , 2009a]Li, B., Yang, Q., and Xue, X.: Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In IJCAI, pages 2052–2057, 2009.

[Li et al. , 2009b]Li, B., Yang, Q., and Xue, X.: Transfer learning for collaborative filtering via a rating-matrix generative model. In ICML, pages 617–624, 2009.

[Li and Lin, 2014]Li, C. and Lin, S.: Matching users and items across domains to improve the recommendation quality. In SIGKDD, pages 801–810, 2014.

[Li et al. , 2011]Li, F., Liu, N. N., Jin, H., Zhao, K., Yang, Q., and Zhu, X.: Incorporating reviewer and product information for review rating prediction. In IJCAI, pages 1820–1825, 2011.

[Li et al. , 2016]Li, Q., Wang, J., Wang, F., Li, P., Liu, L., and Chen, Y.: The role of social sentiment in stock markets: a view from joint effects of multiple information sources. Multimedia Tools and Applications, pages 1–31, 2016.

[Liang et al. , 2017]Liang, T., He, L., Lu, C.-T., Chen, L., Yu, P. S., and Wu, J.: A broad learning approach for context-aware mobile application recommendation. In ICDM, pages 955–960, Nov. 2017.

[Liben-Nowell and Kleinberg, 2003]Liben-Nowell, D. and Kleinberg, J. M.: The link prediction problem for social networks. In CIKM, pages 556–559, 2003.

[Lichtenwalter et al. , 2010]Lichtenwalter, R., Lussier, J. T., and Chawla, N. V.: New perspectives and methods in link prediction. In SIGKDD, pages 243–252, 2010.

[Lin et al. , 2011]Lin, Z., Liu, R., and Su, Z.: Linearized alternating direction method with adaptive penalty for low-rank representation. In NIPS, pages 612–620, 2011.

[Ling et al. , 2014]Ling, G., Lyu, M. R., and King, I.: Ratings meet reviews, a combined approach to recommend. In RecSys, pages 105–112. ACM, 2014.

[Ling et al. , 2008]Ling, X., Dai, W., Xue, G., Yang, Q., and Yu, Y.: Spectral domain-transfer learning. In SIGKDD, pages 488–496, 2008.

[Liu et al. , 2013]Liu, J., Zhang, F., Song, X., Song, Y.-I., Lin, C.-Y., and Hon, H.-W.: What's in a name?: an unsupervised approach to link users across communities. In WSDM, pages 495–504, 2013.

[Livni et al. , 2014]Livni, R., Shalev-Shwartz, S., and Shamir, O.: On the computational efficiency of training neural networks. In NIPS, pages 855–863, 2014.

[Long et al. , 2010]Long, M., Cheng, W., Jin, X., Wang, J., and Shen, D.: Transfer learning via cluster correspondence inference. In ICDM, pages 917–922, 2010.

[Long et al. , 2014]Long, M., Wang, J., Ding, G., Shen, D., and Yang, Q.: Transfer learning with graph co-regularization. TKDE, 26(7):1805–1818, 2014.

[Lu et al. , 2018]Lu, C.-T., He, L., Ding, H., Cao, B., and Yu, P. S.: Learning from multi-view multi-way data via structural factorization machines. In WWW, 2018.

[Lu et al. , 2017]Lu, C.-T., He, L., Shao, W., Cao, B., and Yu, P. S.: Multilinear factorization machines for multi-task multi-view learning. In WSDM, pages 701–709. ACM, 2017.

[Lu et al. , 2014a]Lu, C., Shuai, H., and Yu, P. S.: Identifying your customers in social networks. In CIKM, pages 391–400, 2014.

[Lu et al. , 2014b]Lu, C.-T., Xie, S., Kong, X., and Yu, P. S.: Inferring the impacts of social media on crowdfunding. In WSDM, pages 573–582, 2014.

[Lu et al. , 2016]Lu, C.-T., Xie, S., Shao, W., He, L., and Yu, P. S.: Item recommendation for emerging online businesses. In IJCAI, pages 3797–3803, 2016.

[Lu and Lakshmanan, 2012]Lu, W. and Lakshmanan, L. V. S.: Profit maximization over social networks. In ICDM, pages 479–488, 2012.

[Lu et al. , 2010]Lu, Z., Savas, B., Tang, W., and Dhillon, I. S.: Supervised link prediction using multiple sources. In ICDM, pages 923–928, 2010.

[Luo et al. , 2014]Luo, C., Pang, W., Wang, Z., and Lin, C.: Hete-cf: Social-based collaborative filtering recommendation using heterogeneous relations. In ICDM, pages 917–922, 2014.

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

[Ma et al. , 2011]Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I.: Recommender systems with social regularization. In WSDM, pages 287–296. ACM, 2011.

[Madani et al. , 2013]Madani, O., Georg, M., and Ross, D.: On using nearly-independent feature families for high precision and confidence. Machine learning, 92(2-3):457–477, 2013.

[Malhotra et al. , 2012]Malhotra, A., Totti, L. C., Jr., W. M., Kumaraguru, P., and Almeida, V.: Studying user footprints in different online social networks. In ASONAM, pages 1065–1070, 2012.

[Marie and Gal, 2007]Marie, A. and Gal, A.: On the stable marriage of maximum weight royal couples. In IIWeb, 2007.

[McAuley and Leskovec, 2013]McAuley, J. and Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In RecSys, pages 165–172. ACM, 2013.

[McAuley et al. , 2015]McAuley, J., Pandey, R., and Leskovec, J.: Inferring networks of substitutable and complementary products. In SIGKDD, pages 785–794, 2015.

[McAuley and Leskovec, 2013]McAuley, J. J. and Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In RecSys, pages 165–172, 2013.

[McPherson et al. , 2001]McPherson, M., Lovin, L. S., and Cook, J. M.: Birds of a Feather: Homophily in Social Networks. Annual Review of Sociology, 27(1):415–444, 2001.

[Mebane-Sims, 2009]Mebane-Sims, I.: 2009 alzheimer's disease facts and figures. Alzheimer's & Dementia, 2009.

[Nie et al. , 2010]Nie, F., Huang, H., Cai, X., and Ding, C. H.: Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In NIPS, pages 1813–1821, 2010.

[Noel et al. , 2012]Noel, J., Sanner, S., Tran, K., Christen, P., Xie, L., Bonilla, E. V., Abbasnejad, E., and Penna, N. D.: New objective functions for social collaborative filtering. In WWW, pages 859–868, 2012.

[Novikov et al. , 2017]Novikov, A., Trofimov, M., and Oseledets, I.: Exponential machines. In ICLR, 2017.

[Obozinski et al. , 2010]Obozinski, G., Taskar, B., and Jordan, M. I.: Joint covariate selection and joint subspace selection for multiple classification problems. Statistics and Computing, 20(2):231–252, 2010.

[O'Dell et al. , 1995]O'Dell, M. W., Lubeck, D. P., O'Driscoll, P., and Matsuno, S.: Validity of the karnofsky performance status in an HIV-infected sample. Journal of Acquired Immune Deficiency Syndromes, 10(3):350–357, 1995.

[Okkan and Fistikoglu, 2013]Okkan, U. and Fistikoglu, O.: Evaluating climate change effects on runoff by statistical downscaling and hydrological model gr2m. Theoretical and Applied Climatology, pages 1–19, 2013.

[Pan et al. , 2010]Pan, W., Xiang, E. W., Liu, N. N., and Yang, Q.: Transfer learning in collaborative filtering for sparsity reduction. In AAAI, 2010.

[Peng et al. , 2005]Peng, H., Long, F., and Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. TPAMI, 27(8):1226–1238, 2005.

[Price et al. , 2007]Price, R., Epstein, L., Becker, J., Cinque, P., Gisslén, M., Pulliam, L., and McArthur, J.: Biomarkers of HIV-1 CNS infection and injury. Neurology, 69(18):1781–1788, 2007.

[Qian and Zhai, 2014]Qian, M. and Zhai, C.: Unsupervised feature selection for multi-view clustering on text-image web news data. In CIKM, pages 1963–1966, 2014.

[Qu et al. , 2016]Qu, Y., Cai, H., Ren, K., Zhang, W., Yu, Y., Wen, Y., and Wang, J.: Product-based neural networks for user response prediction. In ICDM, pages 1149–1154. IEEE, 2016.

[Raad et al. , 2010]Raad, E., Chbeir, R., and Dipanda, A.: User profile matching in social networks. In NBiS, pages 297–304, 2010.

[Ragin et al. , 2012]Ragin, A. B., Du, H., Ochs, R., Wu, Y., Sammet, C. L., Shoukry, A., and Epstein, L. G.: Structural brain alterations can be detected early in HIV infection. Neurology, 79(24):2328–2334, 2012.

[Rendle, 2010]Rendle, S.: Factorization machines. In ICDM, pages 995–1000, 2010.

[Rendle, 2012]Rendle, S.: Factorization machines with libfm. ACM TIST, 3(3):57, 2012.

[Rendle, 2013]Rendle, S.: Scaling factorization machines to relational data. In VLDB Endowment, volume 6, pages 337–348. VLDB Endowment, 2013.

[Rendle et al. , 2016]Rendle, S., Fetterly, D., Shekita, E. J., and Su, B.-y.: Robust large-scale machine learning in the cloud. In SIGKDD, pages 1125–1134. ACM, 2016.

[Rendle and Schmidt-Thieme, 2010]Rendle, S. and Schmidt-Thieme, L.: Pairwise interaction tensor factorization for personalized tag recommendation. In WSDM, pages 81–90, 2010.

[Robnik-Šikonja and Kononenko, 2003]Robnik-Šikonja, M. and Kononenko, I.: Theoretical and empirical analysis of relieff and rrelieff. Machine learning, 53(1-2):23–69, 2003.

[Rong et al. , 2014]Rong, Y., Wen, X., and Cheng, H.: A monte carlo algorithm for cold start recommendation. In WWW, pages 327–336, 2014.

[Salton and McGill, 1986]Salton, G. and McGill, M. J.: Introduction to Modern Information Retrieval. New York, NY, USA, McGraw-Hill, Inc., 1986.

[Sarwar et al. , 2001]Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J.: Item-based collaborative filtering recommendation algorithms. In WWW, pages 285–295, 2001.

[Schölkopf et al. , 2001]Schölkopf, B., Herbrich, R., and Smola, A. J.: A generalized representer theorem. In COLT, pages 416–426. Springer, 2001.

[Shan et al. , 2016]Shan, Y., Hoens, T. R., Jiao, J., Wang, H., Yu, D., and Mao, J.: Deep crossing: Web-scale modeling without manually crafted combinatorial features. In SIGKDD, pages 255–262. ACM, 2016.

[Shi et al. , 2014]Shi, C., Kong, X., Huang, Y., Yu, P. S., and Wu, B.: Hetesim: A general framework for relevance measure in heterogeneous networks. TKDE, 26(10):2479–2492, 2014.

[Shi et al. , 2015]Shi, C., Zhang, Z., Luo, P., Yu, P. S., Yue, Y., and Wu, B.: Semantic path based personalized recommendation on weighted heterogeneous information networks. In CIKM, pages 453–462. ACM, 2015.

[Shi et al. , 2012]Shi, C., Zhou, C., Kong, X., Yu, P. S., Liu, G., and Wang, B.: Heterecom: a semantic-based recommendation systemin heterogeneous networks. In SIGKDD, pages 1552–1555, 2012.

[Signoretto, 2011]Signoretto, M.: Kernels and tensors for structured data modelling. Doctoral dissertation, 2011.

[Sindhwani and Rosenberg, 2008]Sindhwani, V. and Rosenberg, D. S.: An rkhs for multi-view learning and manifold co-regularization. In ICML, pages 976–983, 2008.

[Singh and Gordon, 2008]Singh, A. P. and Gordon, G. J.: Relational learning via collective matrix factorization. In SIGKDD, pages 650–658. ACM, 2008.

[Smalter et al. , 2009]Smalter, A., Huan, J., and Lushington, G.: Feature selection in the tensor product feature space. In ICDM, pages 1004–1009, 2009.

[Sun, 2013]Sun, S.: A survey of multi-view machine learning. Neural Computing and Applications, 23(7-8):2031–2038, 2013.

[Sun et al. , 2011]Sun, Y., Han, J., Yan, X., Yu, P. S., and Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. PVLDB, 4(11):992–1003, 2011.

[Tang et al. , 2012]Tang, J.and Gao, H., Liu, H., and Das Sarma, A.: eTrust: Understanding trust evolution in an online world. In SIGKDD, pages 253–261. ACM, 2012.

[Tang et al. , 2013]Tang, J., Hu, X., Gao, H., and Liu, H.: Unsupervised feature selection for multi-view data in social media. In SDM, 2013.

[Tao et al. , 2005]Tao, D., Li, X., Hu, W., Maybank, S., and Wu, X.: Supervised tensor learning. In ICDM, pages 8–pp, 2005.

[Tao et al. , 2007]Tao, D., Li, X., Wu, X., Hu, W., and Maybank, S. J.: Supervised tensor learning. KAIS, 13(1):1–42, 2007.

[Tsytsarau et al. , 2013]Tsytsarau, M., Amer-Yahia, S., and Palpanas, T.: Efficient sentiment correlation for large-scale demographics. In SIGMOD, pages 253–264, 2013.

[Vahedian, 2014]Vahedian, F.: Weighted hybrid recommendation for heterogeneous networks. In RecSys, pages 429–432. ACM, 2014.

[Vapnik, 2013]Vapnik, V.: The nature of statistical learning theory. Springer Science & Business Media, 2013.

[Varma and Babu, 2009]Varma, M. and Babu, B. R.: More generality in efficient multiple kernel learning. In ICML, pages 1065–1072, 2009.

[Wang and Mahadevan, 2011]Wang, C. and Mahadevan, S.: Heterogeneous domain adaptation using manifold alignment. In IJCAI, pages 1541–1546, 2011.

[Wang et al. , 2007]Wang, C., Satuluri, V., and Parthasarathy, S.: Local probabilistic models for link prediction. In ICDM, pages 322–331, 2007.

[Wang et al. , 2013a]Wang, H., Nie, F., and Huang, H.: Multi-view clustering and feature learning via structured sparsity. In ICML, pages 352–360, 2013.

[Wang et al. , 2013b]Wang, H., Nie, F., Huang, H., and Ding, C.: Heterogeneous visual features fusion via sparse multimodal machine. In CVPR, pages 3097–3102, 2013.

[Wang et al. , 2017]Wang, R., Fu, B., Fu, G., and Wang, M.: Deep & cross network for ad click predictions. arXiv preprint arXiv:1708.05123, 2017.

[Wang et al. , 2009]Wang, Z., Song, Y., and Zhang, C.: Knowledge transfer on hybrid graph. In IJCAI, pages 1291–1296, 2009.

[Weis et al. , 1993]Weis, S., Haug, H., and Budka, H.: Neuronal damage in the cerebral cortex of AIDS brains: a morphometric study. Acta neuropathologica, 85(2):185–189, 1993.

[White et al. , 2012]White, M., Zhang, X., Schuurmans, D., and Yu, Y.-l.: Convex multi-view subspace learning. In NIPS, pages 1673–1681, 2012.

[Xiao et al. , 2017]Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., and Chua, T.-S.: Attentional factorization machines: Learning the weight of feature interactions via attention networks. In IJCAI, 2017.

[Xie et al. , 2014]Xie, C., Yan, L., Li, W.-J., and Zhang, Z.: Distributed power-law graph computing: Theoretical and empirical analysis. In NIPS, pages 1673–1681, 2014.

[Xu et al. , 2013]Xu, C., Tao, D., and Xu, C.: A survey on multi-view learning. arXiv:1304.5634, 2013.

[Yan et al. , 2000]Yan, J., Blackwell, A., Anderson, R., and Grant, A.: The memorability and security of passwords – some empirical results. Technical report, University of Cambridge, Computer Laboratory, Sep 2000.

[Yan et al. , 2014]Yan, L., Li, W.-j., Xue, G.-R., and Han, D.: Coupled group lasso for web-scale CTR prediction in display advertising. In ICML, pages 802–810, 2014.

[Yang and He, 2014]Yang, H. and He, J.: Learning with dual heterogeneity: a nonparametric bayes model. In SIGKDD, pages 582–590. ACM, 2014.

[Yang et al. , 2014]Yang, X., Guo, Y., Liu, Y., and Steck, H.: A survey of collaborative filtering based social recommender systems. Computer Communications, 41:1 – 10, 2014.

[Ye et al. , 2008]Ye, J., Chen, K., Wu, T., Li, J., Zhao, Z., Patel, R., Bae, M., Janardan, R., Liu, H., Alexander, G., and Reiman, E.: Heterogeneous data fusion for Alzheimer's disease study. In SIGKDD, pages 1025–1033, 2008.

[Ye et al. , 2012]Ye, M., Liu, X., and Lee, W.-C.: Exploring social influence for recommendation: a generative model approach. In SIGIR, pages 671–680, 2012.

[Yu et al. , 2013]Yu, X., Ren, X., Gu, Q., Sun, Y., and Han, J.: Collaborative filtering with entity similarity regularization in heterogeneous information networks. In IJCAI HINA, 2013.

[Yu et al. , 2014]Yu, X., Ren, X., Sun, Y., Gu, Q., Sturt, B., Khandelwal, U., Norick, B., and Han, J.: Personalized entity recommendation: a heterogeneous information network approach. In WSDM, pages 283–292, 2014.

[Yu et al. , 2013]Yu, X., Ren, X., Sun, Y., Sturt, B., Khandelwal, U., Gu, Q., Norick, B., and Han, J.: Recommendation in heterogeneous information networks with implicit user feedback. In RecSys, pages 347–350, 2013.

[Yuan et al. , 2011]Yuan, Q., Chen, L., and Zhao, S.: Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In RecSys, pages 245–252, 2011.

[Zadrozny and Elkan, 2002]Zadrozny, B. and Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In SIGKDD, pages 694–699, 2002.

[Zafarani and Liu, 2013]Zafarani, R. and Liu, H.: Connecting users across social media sites: a behavioral-modeling approach. In SIGKDD, pages 41–49, 2013.

[Zaharia et al. , 2012]Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M. J., Shenker, S., and Stoica, I.: Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In NSDI, pages 2–2. USENIX, 2012.

[Zhang et al. , 2015]Zhang, J., Shao, W., Wang, S., Kong, X., and Yu, P. S.: PNA: Partial network alignment with generic stable matching. In IRI, pages 166–173. IEEE, 2015.

[Zhang and Yu, 2015]Zhang, J. and Yu, P. S.: Community detection for emerging networks. In SDM. SIAM, 2015.

[Zhang et al. , 2014]Zhang, J., Yu, P. S., and Zhou, Z.: Meta-path based multi-network collective link prediction. In SIGKDD, pages 1286–1295, 2014.

[Zhang et al. , 2017]Zhang, J., Lu, C.-T., Cao, B., Chang, Y., and Yu, P. S.: Connecting emerging relationships from news via tensor factorization. In Big Data. IEEE, 2017.

[Zhang et al. , 2016]Zhang, J., Lu, C.-T., Zhou, M., Xie, S., Chang, Y., and Philip, S. Y.: Heer: Heterogeneous graph embedding for emerging relation detection from news. In Big Data, pages 803–812. IEEE, 2016.

[Zhang and Huan, 2012]Zhang, J. and Huan, J.: Inductive multi-task learning with multiple view data. In SIGKDD, pages 543–551, 2012.

[Zhang et al. , 2006]Zhang, S., Wang, W., Ford, J., and Makedon, F.: Learning from incomplete ratings using non-negative matrix factorization. In SDM, pages 549–553, 2006.

[Zhang et al. , 2016]Zhang, W., Du, T., and Wang, J.: Deep learning over multi-field categorical data. In ECIR, pages 45–57. Springer, 2016.

[Zhang and Pennacchiotti, 2013]Zhang, Y. and Pennacchiotti, M.: Predicting purchase behaviors from social media. In WWW, pages 1521–1532, 2013.

[Zhang and Yeung, 2012]Zhang, Y. and Yeung, D.-Y.: A convex formulation for learning task relationships in multi-task learning. arXiv preprint arXiv:1203.3536, 2012.

[Zhang and Tang, 2013]Zhang, Y. and Tang, J.: Social network integration: Towards constructing the social graph. CoRR, abs/1311.2670, 2013.

[Zheng et al. , 2017]Zheng, L., Noroozi, V., and Yu, P. S.: Joint deep modeling of users and items using reviews for recommendation. In WSDM, pages 425–434. ACM, 2017.

[Zhou et al. , 2017]Zhou, G., Song, C., Zhu, X., Ma, X., Yan, Y., Dai, X., Zhu, H., Jin, J., Li, H., and Gai, K.: Deep interest network for click-through rate prediction. arXiv preprint arXiv:1706.06978, 2017.

[Zhou et al. , 2011]Zhou, J., Chen, J., and Ye, J.: Clustered multi-task learning via alternating structure optimization. In NIPS, pages 702–710, 2011.

[Zhou et al. , 2009]Zhou, T., Lü, L., and Zhang, Y.-C.: Predicting missing links via local information. The European Physical Journal B - Condensed Matter and Complex Systems, 71(4):623–630, October 2009.

[Zhu et al. , 2017]Zhu, J., Shan, Y., Mao, J., Yu, D., Rahmanian, H., and Zhang, Y.: Deep embedding forest: Forest-based serving with deep embedding features. In SIGKDD, 2017.

[Zhu and Lafferty, 2005]Zhu, X. and Lafferty, J. D.: Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In ICML, pages 1052–1059, 2005.

[Zhu et al. , 2010]Zhu, Y., Chen, Y., Lu, Z., Pan, S. J., rong Xue, G., Yu, Y., and Yang, Q.: Heterogeneous transfer learning for image classification. In AAAI, 2010.

VITA

Name: Chun-Ta Lu

EDUCATION:

M.S. in Computer Science, National Chiao Tung University, 2011.

B.S. in Computer Science, National Taiwan University, 2008.

PUBLICATIONS:

- Chun-Ta Lu, Lifang He, Hao Ding, Bokai Cao, and Philip S. Yu. *Learning from Multi-View Multi-Way Data via Structural Factorization Machines.* In The Web Conference **(WWW)**, 2018.
- Chun-Ta Lu, Lifang He, Weixiang Shao, Bokai Cao, and Philip S. Yu. *Multilinear Factorization Machines for Multi-Task Multi-View Learning.* In ACM International Conference on Web Search and Data Mining **(WSDM)**, 2017.
- Lifang He, Chun-Ta Lu, Guixiang Ma, Shen Wang, Linlin Shen, Philip S. Yu, and Ann B. Ragin. *Kernelized Support Tensor Machines.* In International Conference on Machine Learning **(ICML)**, 2017.
- Lifang He, Chun-Ta Lu, Hao Ding, Shen Wang, Linlin Shen, Philip S. Yu, and Ann B. Ragin. *Multi-way Multi-level Kerenl Modeling for Neuroimaging Classification.* In IEEE Conference on Computer Vision and Pattern Recognition **(CVPR)**, 2017.

- Shen Wang, Lifang He, Bokai Cao, Chun-Ta Lu, Philip S Yu and Ann B. Ragin. *Structural Deep Brain Network Mining.* In ACM SIGKDD Conference on Knowledge Discovery and Data Mining **(KDD)**, 2017 (Oral).
- Tingting Liang, Lifang He, Chun-Ta Lu, Liang Chen, Philip S. Yu and Jian Wu. *A Broad Learning Approach for Context-Aware Mobile Application Recommendation.* In IEEE International Conference on Data Mining **(ICDM)**, 2017.
- Guixiang Ma, Chun-Ta Lu, Lifang He, Philip S. Yu, Ann Ragin. *Multi-view Graph Embedding with Hub Detection for Brain Network Analysis.* In IEEE International Conference on Data Mining **(ICDM)**, 2017.
- Guixiang Ma, Lifang He, Chun-Ta Lu, Weixiang Shao, Philip S Yu, Alex D Leow, and Ann B Ragin. *Multi-view clustering with graph embedding for connectome analysis.* In ACM International Conference on Information and Knowledge Management **(CIKM)**, 2017.
- Fengjiao Wang, Chun-Ta Lu, Yongzhi Qu, and Philip S. Yu. *Collective Geographical Embedding for Geolocating Social Network Users*, In Pacific Asia Conference on Knowledge Discovery and Data Mining **(PAKDD)**, 2017.
- Fengjiao Wang, Yongzhi Qu, Lei Zheng, Chun-Ta Lu, and Philip S Yu. *Deep and Broad Learning on Content-Aware POI Recommendation.* In International Conference on Collaboration and Internet Computing **(CIC)**, 2017.

- Jingyuan Zhang, Chun-Ta Lu, Bokai Cao, Yi Chang, Philip S. Yu. *Connecting Emerging Relationships from News via Tensor Factorization.* In IEEE International Conference on Big Data **(IEEE BigData)**, 2017.

- Chenwei Zhang, Nan Du, Wei Fan, Yaliang Li, Chun-Ta Lu, Philip S. Yu. *Bringing Semantic Structures to User Intent Detection in Online Medical Queries.* In IEEE International Conference on Big Data **(IEEE BigData)**, 2017.

- Chun-Ta Lu, Sihong Xie, Weixiang Shao, Lifang He, and Philip S. Yu. *Item Recommendation for Emerging Online Businesses.* In International Joint Conference on Artificial Intelligence **(IJCAI)**, 2016.

- Lifang He, Chun-Ta Lu, Jiaqi Ma, Jianping Cao, Linlin Shen, and Philip S. Yu. *Joint Community and Structural Hole Spanner Detection via Harmonic Modularity.* In ACM SIGKDD Conference on Knowledge Discovery and Data Mining **(KDD)**, 2016 (Oral).

- Bokai Cao, Chun-Ta Lu, Xiaokai Wei, Philip S. Yu, and Alex D. Leow. *Semi-supervised Tensor Factorization for Brain Network Analysis*, In Joint European Conference on Machine Learning and Knowledge Discovery in Databases **(ECML/PKDD)**, 2016.

- Weixiang Shao, Lifang He, Chun-Ta Lu, Xiaokai Wei and Philip S. Yu. *Online Unsupervised Multi-view Feature Selection.* In IEEE International Conference on Data Mining **(ICDM)**, 2016.

- Guixiang Ma, Lifang He, Chun-Ta Lu, Philip S. Yu, Linlin Shen, and Ann B. Ragin. *Spatio-Temporal Tensor Analysis for Whole-Brain fMRI Classification*, In SIAM International Conference on Data Mining **(SDM)**, 2016.

- Jingyuan Zhang, Bokai Cao, Sihong Xie, Chun-Ta Lu, Philip S. Yu, and Ann B. Ragin. *Identifying Connectivity Patterns for Brain Diseases via Multi-side-view Guided Deep Architectures*, In SIAM International Conference on Data Mining **(SDM)**, 2016.

- Jingyuan Zhang, Chun-Ta Lu, Mianwei Zhou, Sihong Xie, Yi Chang, and Philip S. Yu. *HEER: Heterogeneous Graph Embedding for Emerging Relation Detection from News*, In IEEE International Conference on Big Data **(IEEE BigData)**, 2016.

- Weixiang Shao, Lifang He, Chun-Ta Lu and Philip S. Yu. *Online Multi-view Clustering with Incomplete Views*, In IEEE International Conference on Big Data **(IEEE BigData)**, 2016.

- Yuan Yuan, Sihong Xie, Chun-Ta Lu, Philip S. Yu, and Jie Tang. *Interpretable and Effective Opinion Spam Detection via Temporal Patterns Mining across Websites*, In IEEE International Conference on Big Data **(IEEE BigData)**, 2016.

- Xiaokai Wei, Bokai Cao, Weixiang Shao, Chun-Ta Lu and Philip S. Yu. *Community Detection with Partially Observable Links and Node Attributes*, In IEEE International Conference on Big Data **(IEEE BigData)**, 2016.

- Chun-Ta Lu, Hong-Han Shuai, and Philip S. Yu. *Identifying your Customers in Social Networks.* In ACM International Conference on Information and Knowledge Management **(CIKM)**, 2014.
- Chun-Ta Lu, Sihong Xie, Xiangnan Kong, and Philip S. Yu. *Inferring the Impacts of Social Media on Crowdfunding.* In ACM International Conference on Web Search and Data Mining **(WSDM)**, 2014.
- Chun-Ta Lu, Po-Ruey Lei, Wen-Chih Peng, and Ing-Jiunn Su. *A Framework for Mining Semantic Regions from Trajectories*, In International Conference on Database Systems for Advanced Application **(DASFAA)**, 2011.