Data Min Knowl Disc
DOI 10.1007/s10618-013-0341-y

Link prediction in heterogeneous data via generalized coupled tensor factorization

Beyza Ermiş · Evrim Acar · A. Taylan Cemgil

Received: 29 December 2012 / Accepted: 2 December 2013

Abstract This study deals with missing link prediction, the problem of predicting the existence of missing connections between entities of interest. We approach the problem as filling in missing entries in a relational dataset represented by several matrices and multiway arrays, that will be simply called *tensors*. Consequently, we address the link prediction problem by data fusion formulated as simultaneous factorization of several observation tensors where latent factors are shared among each observation. Previous studies on joint factorization of such heterogeneous datasets have focused on a single loss function (mainly squared Euclidean distance or Kullback–Leibler-divergence) and specific tensor factorization models (CANDECOMP/PARAFAC and/or Tucker). However, in this paper, we study various alternative tensor models as well as loss functions including the ones already studied in the literature using the generalized coupled tensor factorization framework. Through extensive experiments on two real-world datasets, we demonstrate that (i) joint analysis of data from multiple sources via coupled factorization significantly improves the link prediction performance, (ii) selection of a suitable loss function and a tensor factorization model is crucial for accurate missing link prediction and loss functions that have not been studied for link prediction before may outperform the commonly-used loss functions, (iii) joint factorization of datasets can handle difficult cases, such as the *cold start problem*

Responsible editor: Jian Pei.

B. Ermiş (✉) · A. T. Cemgil
Department of Computer Science, Boğaziçi University, Bebek, 34342 Istanbul, Turkey
e-mail: beyza.ermis@boun.edu.tr; ermisbeyza@gmail.com

A. T. Cemgil
e-mail: taylan.cemgil@boun.edu.tr

E. Acar
Faculty of Life Sciences, University of Copenhagen, 1958 Frederiksberg C, Denmark
e-mail: evrim@life.ku.dk

Published online: 24 December 2013

that arises when a new entity enters the dataset, and (iv) our approach is scalable to large-scale data.

Keywords Coupled tensor factorization · Link prediction · Heterogeneous data · Missing data · Data fusion

1 Introduction

Recent technological advances, such as the Internet, multi-media devices or social networks provide abundance of relational data. For instance, in retail recommender systems, typically a retailer will have access to retail data showing *who has bought which items*, we may also have access to customers' social networks, i.e., *who is friends with whom*. Clearly, the social network data may provide valuable side information and jointly analyzing data from multiple sources has great potential to increase our ability for accurate prediction of missing data. In this study, we focus on a particular task for relational data modeling: *link prediction*.

Applications in many areas including recommender systems and social network analysis deal with link prediction, i.e., the problem of inferring whether there is a relation between the entities of interest. For instance, if a customer buys an item, the customer and the item can be considered to be linked. The task of recommending other items the customer may be interested in can be cast as a missing link prediction problem. However, the results are likely to be poor if the prediction is done in isolation on a single view of data. Such datasets, whilst large in dimension, are already very sparse (Getoor and Diehl 2005) and potentially represent only a very incomplete picture of the reality (Clauset et al. 2008). Therefore, relational data from other sources is often incorporated into link prediction models (Cao et al. 2010; Davis et al. 2011; Menon and Elkan 2011; Popescul and Ungar 2003; Taskar et al. 2003; Yang et al. 2011, 2012).

An effective way of including side information via additional relational data in a link prediction model is to represent different relations as a collection of matrices. Subsequently, this collection of matrices are jointly analyzed using collective matrix factorization, CMF (Long et al. 2006; Singh and Gordon 2008). Joint factorization of matrices have proved useful in many social networking applications (Jiang et al. 2012; Koren et al. 2009; Ma et al. 2008; Menon et al. 2011; Yang et al. 2011; Yoo and Choi 2012). However, matrices are often not sufficient for a faithful representation of multiple attributes, and higher-order tensor and matrix factorization models are needed. An influential study in this direction is by Banerjee et al. (2007), where a general clustering method for joint analysis of heterogeneous data has been studied. The goal here is clustering entities based on multiple relations, where each relation is represented as a matrix (e.g., movies by review words matrix showing movie reviews) or a higher-order tensor (e.g., movies by viewers by actors tensor showing viewers' ratings).

Various algorithms have been proposed in the literature for coupled analysis of heterogeneous data. Lin et al. (2009) propose a factorization method for community extraction on multi-relational and multi-dimensional social data by using relational

hypergraph representation. Their coupled factorization approach models higher-order tensors using a specific tensor model, i.e., CANDECOMP/PARAFAC (CP) (Carroll and Chang 1970; Harshman 1970; Hitchcock 1927), and has a Kullback–Leibler (KL) divergence-based cost function. Also, a recent study by Narita et al. (2011) has considered joint factorization of coupled matrices and higher-order tensors based on CP and Tucker (1963, 1966) models using a Euclidean (EUC) distance-based loss function.

In this article, we address link prediction problem using coupled analysis of datasets in the form of matrices and higher-order tensors. Unlike previous studies on coupled analysis of heterogeneous datasets focusing on a certain loss function or a specific tensor model, we use an approach, i.e., generalized coupled tensor factorizations, GCTFs (Yilmaz et al. 2011), based on a probabilistic interpretation of tensor factorization models as generalized linear models, which enables us to investigate alternative tensor models and cost functions in addition to the approaches already studied in the literature. Table 1 shows the related work in coupled factorizations, which can all be considered as special cases of the GCTF framework[1] in terms of the loss functions and tensor models they consider. We assess the performance of those related studies as special cases of GCTF (and baseline methods) in our experiments. The main contributions of this article can be summarized as follows:

- Addressing link prediction using joint analysis of heterogeneous data based on different tensor models, i.e., CP, Tucker and some arbitrary tensor factorization models, as well as different loss functions, i.e., KL-divergence, IS (Itakura–Saito)-divergence, EUC distance and various other cost functions based on β-divergences.
- Demonstrating on two real datasets that coupled tensor factorizations outperform low-rank approximations of a single tensor and the selection of the tensor model as well as the loss function is significant in terms of link prediction performance.
- Handling the cold-start problem in link prediction using the proposed models accurately.
- Demonstrating the scalability of the proposed models on a large-scale dataset.

This is an extended version of our previous study (Ermis et al. 2012), where we have used the GCTF framework for link prediction but only considering CP and Tucker models using EUC distance and KL-divergence based loss functions on a small dataset. In this paper, we assess the performance of arbitrary tensor factorization models and various cost functions based on β-divergences (including IS-divergence) in order to demonstrate the flexibility of the GCTF framework for the link prediction problem. Numerical experiments demonstrate that loss functions that have not been studied for link prediction before, such as IS, may outperform the commonly-used loss functions. Therefore, it is extremely useful to explore alternative loss functions using the GCTF framework for the link prediction problem for different datasets. Furthermore, we also show the scalability of our approach on a large-scale real dataset.

The rest of the article is organized as follows. In Sect. 2, we survey the related work on link prediction as well as joint factorization of data. Section 3 introduces

[1] Some of the listed studies do not impose nonnegativity constraints on the factor matrices while GCTF assumes that all factor matrices are nonnegative.

our algorithmic framework, i.e., GCTF, while Sect. 4 discusses its adaptation for the link prediction problem. Experimental results on real datasets are presented in Sect. 5. Finally, we conclude in Sect. 6.

2 Related work

In order to deal with the challenging task of link prediction, many studies have proposed to exploit multi-relational nature of the data and showed improved link prediction performance by incorporating related sources of information in their modeling framework. For instance, earlier work by Taskar et al. (2003) uses relational Markov networks to model links between entities as well as their attributes. Popescul and Ungan (2003) extract relational features to learn the existence of links (see Al Hasan and Zaki 2011 for a comprehensive list of similar studies). More recently, Cao et al. (2010) have proposed a nonparametric Bayesian framework for collective link prediction by developing a multitask extension of the Gaussian-process latent variable model. Also, Davis et al. (2011) explore triad information in heterogeneous networks while Yang et al. (2012) use a new topological feature to capture the correlations between different types of links for the link prediction problem.

For analysis of multi-relational data, Singh and Gordon (2008) as well as Long et al. (2006) have introduced CMFs. Matrix factorization-based techniques have proved useful in terms of capturing the underlying patterns in data, e.g., in collaborative filtering (Koren et al. 2009; Menon et al. 2011), and joint analysis of matrices has been widely applied in numerous disciplines including signal processing (Yoo et al. 2010), bioinformatics (Alter et al. 2003) and social network analysis (Jiang et al. 2012; Koren et al. 2009; Ma et al. 2008; Yang et al. 2011; Yoo and Choi 2012). For instance, Ma et al. (2008) propose a method based on probabilistic factor analysis to make *social recommendation* by integrating social network structure and the user–item rating matrix. They fuse these two different data resources through the shared user latent feature space. Also, Yoo and Choi (2012) extend such CMF models to a Bayesian matrix co-factorization model to exploit side information, e.g., content information and user demographic data, into collaborative prediction problem by using a variational inference algorithm. Besides, Yang et al. (2011) use a coupled latent factor model with variety of differentiable loss functions to uncover missing links.

Recent studies extend CMF to coupled analysis of multi-relational data in the form of matrices and higher-order tensors (Banerjee et al. 2007; Smilde et al. 2000) since in many disciplines, relations can be defined among more than two entities, e.g., when a user engages in an activity at a certain location, a relation can be defined over user, activity and location entities. For instance, Zheng et al. (2012) model the user–location–activity relations with a tensor representation, and propose a matrix and tensor decomposition solution for collaborative location and activity filtering. Banerjee et al. (2007) introduce a multi-way clustering approach for relational and multi-relational data where coupled analysis of heterogeneous data is studied using minimum Bregman information. Lin et al. (2009) also discuss coupled matrix and tensor factorizations using KL-divergence modeling higher-order tensors by fitting a

Table 1 Related studies on coupled factorization of heterogenous data

Methods	Cost functions			Tensor models	
	EUC	KL	IS	CP	Tucker
PCLAF (Zheng et al. 2010, 2012)	✓			✓	
Metafac (Lin et al. 2009)		✓		✓	
Narita et al. (2011)	✓			✓	✓
Acar et al. (2011a)	✓			✓	

CP model. While these studies use alternating algorithms, Acar et al. (2011a) propose an all-at-once optimization approach for coupled analysis. Table 1 summarizes some of the related work on coupled analysis of heterogeneous data in terms of the loss functions and tensor models they study.

Missing link prediction is also closely related to matrix and tensor completion studies. By using a low-rank structure of a data set, it is possible to recover missing entries for matrices (Candès and Plan 2010) and higher-order tensors (Acar et al. 2011b; Gandy et al. 2011). A recent study by Narita et al. (2011) addresses the tensor completion problem using additional data. Note that, in this article, we do not address the temporal link prediction problem, where snapshots of the set of links up to time t are given and the goal is to predict the links at time $t+1$. Tensor factorizations have previously been used for temporal link prediction (Dunlavy et al. 2011). We keep our focus limited to missing link prediction in this article.

In addition, there are some existing work which compares the factorization-based methods to other link prediction methods in heterogeneous networks. In their work 2011, Menon and Elkan list some popular link prediction approaches and compares these methods. Then, they conclude that factorization models have many advantages for heterogeneous data: the graphs with several thousands of nodes and millions of edges can be trained using stochastic gradient descent and also, the factorization models can be extended to incorporate side information and overcome the imbalance problem. Jamali and Lakshmanan (2013) review some related wok in heterogeneous networks (Shi et al. 2012; Sun et al. 2011; Wang et al. 2011; Yu et al. 2012) and conclude that these methods are slower in prediction and not appropriate to build scalable algorithms compared to model-based approaches such as CMF that do not require to access the raw data after the learning phase. In this article, our main focus is the factorization-based approaches with different models and loss functions.

3 Methodology

In this section, we first briefly discuss β-divergences within the context of tensor factorizations and then explain probabilistic latent tensor factorization, PLTF (Yilmaz and Cemgil 2010) for factorization of a single tensor. Finally, we introduce the GCTF framework (Yilmaz et al. 2011), which is the generalization of PLTF to coupled factorization of multiple tensors.

3.1 β-Divergences

A tensor factorization problem is specified by an observed data tensor X and a collection of latent factors to be estimated to best fit the data, $Z_{1:|\alpha|} = \{Z_\alpha\}$ for $\alpha = 1, \ldots, |\alpha|$. Error minimization between the observation X and the model output $\hat{X}$ is one of the significant methods used for computation of the latent factors. After computation, this error is distributed back proportionally to the factors and they are adjusted accordingly in an iterative update schema (Yilmaz 2012). We use various divergences between the observed data X and model prediction $\hat{X}$ denoted by $D(X \parallel \hat{X})$ to quantify the quality of the approximation. The iterative algorithm, then, optimizes the factors in the direction of the minimum error

$$\hat{X}^* = \underset{\hat{X}}{\operatorname{argmin}} D(X \parallel \hat{X}).$$

In applications, D is typically taken as EUC distance or KL-divergence. On the other hand, GCTF framework is defined for a large family of loss functions called the β-divergences, which generalizes these commonly-used divergences. β-divergences are defined as (Cichocki et al. 2009):

$$d_p(X;\, \hat{X}) = \frac{X^{2-p}}{(1-p)(2-p)} - \frac{X\hat{X}^{1-p}}{1-p} + \frac{\hat{X}^{2-p}}{2-p},$$

where p determines the cost function. Note that $p = \{0,\ 1,\ 2\}$ corresponds to EUC, KL, and IS cost functions, respectively. In Sect. 5, we illustrate why a specific cost function works well in practice by conducting experiments on synthetic datasets. In our experiments, while we mainly focus on the performance of $p = \{0,\ 1,\ 2\}$, we also explore the performance of link prediction models for p-values in [0–2] interval on our second dataset in order to show the effect of p on link prediction.

3.1.1 Estimation of the p parameter

There are existing matrix and tensor factorization algorithms that minimize the β-divergence (Cichocki et al. 2009; Tan and Fevotte 2013; Yilmaz et al. 2011). These algorithms estimate the mean parameter. However, it is possible to estimate a specific β-divergence for a dataset and power parameter p which is useful for choosing a suitable divergence by utilizing the close connection between β-divergences and a particular exponential family, the so-called Tweedie models (Yılmaz and Cemgil 2012). In Simsekli et al. (2013a), they focus on estimating p when $p \in (1,\ 2)$ by using several inference algorithms in any matrix and tensor factorization model and they also working on estimating p for a wider interval ($p = \{0,\ 1,\ 2,\ 3\}$).

3.2 Probabilistic latent tensor factorization

PLTF enables one to incorporate domain specific information to any arbitrary factorization model and provides the update rules for multiplicative gradient descent and

expectation–maximization algorithms. In this framework, the goal is to compute an approximate factorization of X in terms of a product of individual factors Z_α. Here, we define V as the set of all indices in a model, V_0 as the set of visible indices, V_α as the set of indices in Z_α, and $\bar{V}_\alpha = V - V_\alpha$ as the set of all indices not in Z_α. We use small letters as v_α to refer to a particular setting of indices in V_α.

PLTF tries to solve the following approximation problem

$$X(v_0) \approx \hat{X}(v_0) = \sum_{\bar{v}_0} \prod_{\alpha} Z_\alpha(v_\alpha). \tag{1}$$

Since the product $\prod_\alpha Z_\alpha(v_\alpha)$ is collapsed over a set of indices, the factorization is latent. The approximation problem is cast as an *optimization* problem where we minimize the divergence $D(X, \hat{X})$.

In this paper, we use nonnegative variants of the most widely-used low-rank tensor factorization models, i.e., Tucker model and the more restricted CP model for comparison with our coupled models in Sect. 5. These models can be defined in the PLTF notation as follows. Given a three-way tensor X, its CP model is defined as:

$$X(i, j, k) \approx \hat{X}(i, j, k) = \sum_{r} Z_1(i, r) Z_2(j, r) Z_3(k, r), \tag{2}$$

where the index sets $V = \{i, j, k, r\}$, $V_0 = \{i, j, k\}$, $V_1 = \{i, r\}$, $V_2 = \{j, r\}$ and $V_3 = \{k, r\}$. A Tucker model of X is defined in the PLTF notation as follows:

$$X(i, j, k) \approx \hat{X}(i, j, k) = \sum_{p,q,r} Z_1(i, p) Z_2(j, q) Z_3(k, r) Z_4(p, q, r), \tag{3}$$

where the index sets $V = \{i, j, k, p, q, r\}$, $V_0 = \{i, j, k\}$, $V_1 = \{i, p\}$, $V_2 = \{j, q\}$, $V_3 = \{k, r\}$ and $V_4 = \{p, q, r\}$.

The update equation for non-negative generalized tensor factorization can be used for both (2) and (3) and is expressed as (Yilmaz and Cemgil 2010):

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\Delta_\alpha(M \circ \hat{X}^{-p} \circ X)}{\Delta_\alpha(M \circ \hat{X}^{1-p})} \quad \text{s.t.} \quad Z_\alpha(v_\alpha) > 0, \tag{4}$$

where $\circ$ is the Hadamard product (element-wise product), M is a 0–1 mask array with $M(v_0) = 1$ ($M(v_0) = 0$) if $X(v_0)$ is observed (missing). Here p indicates the cost function and remember that $p = \{0, 1, 2\}$ corresponds EUC, KL, and IS cost functions, respectively. In this iteration, we define the tensor valued function $\Delta_\alpha(A)$ as:

$$\Delta_\alpha(A) = \sum_{\bar{v}_\alpha} A(v_0) \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'}), \tag{5}$$

$\Delta_\alpha(A)$ is an object, the same size of Z_α, obtained simply by multiplying all factors other than the one being updated with an object of the order of the data. Hence the key

observation is that the Δ_α function is just computing a tensor product and collapses this product over indices not appearing in Z_α, which is algebraically equivalent to computing a marginal sum.

As an example, for KL cost, we rewrite (4) more compactly as:

$$Z_\alpha \leftarrow Z_\alpha \circ \Delta_\alpha(M \circ X/\hat{X})/\Delta_\alpha(M). \tag{6}$$

This update rule can be used iteratively for all non-negative Z_α and converges to a local minimum provided we start from some non-negative initial values. For updating Z_α, we need to compute the Δ function twice for arguments $A = M_\nu \circ \hat{X}_\nu^{-p} \circ X_\nu$ and $A = M_\nu \circ \hat{X}_\nu^{1-p}$. It is easy to verify that update equations for the KL-non-negative matrix factorization problem (for $p=1$) are obtained as a special case of (4).

Furthermore, we show the multiplicative update rule for the CP model given in Eq. 2 generated by PLTF with KL cost function. The model estimate and the fixed point equation for Z_1 are as follows:

$$Z_1(i,\, r) \leftarrow Z_1(i,\, r) \frac{\sum_{j,k}(M(i,\, j,\, k)X(i,\, j,\, k)/\hat{X}(i,\, j,\, k))Z_2(j,\, r)Z_3(k,\, r)}{\sum_{j,k} M(i,\, j,\, k)Z_2(j,\, r)Z_3(k,\, r)}. \tag{7}$$

As a further example, this rule specializes for the update of Z_4 factor in the Tucker model given in Eq. 3 to

$$\begin{aligned} Z_4(p,\, q,\, r) &\leftarrow Z_4(p,\, q,\, r) \\ &\times \frac{\sum_{i,j,k} Z_1(i,\, p)Z_2(j,\, q)Z_3(k,\, r)M(i,\, j,\, k)\hat{X}(i,\, j,\, k)/X(i,\, j,\, k)}{\sum_{i,j,k} Z_1(i,\, p)Z_2(j,\, q)Z_3(k,\, r)M(i,\, j,\, k)}. \end{aligned} \tag{8}$$

Other factor updates are similar. Note that these updates respect the sparsity pattern of the data X as specified by the mask M and can be efficiently implemented on large-but-sparse data as we illustrate with our experiments in Sect. 5 and Appendix 6.

3.3 Generalized coupled tensor factorization

The GCTF model takes the PLTF model one step further where, in this case, we have multiple observed tensors X_ν that are supposed to be factorized simultaneously:

$$X_\nu\left(v_{0,\nu}\right) \approx \hat{X}_\nu\left(v_{0,\nu}\right) = \sum_{\bar{v}_{0,\nu}} \prod_\alpha Z_\alpha\left(v_\alpha\right)^{R^{\nu,\alpha}}, \tag{9}$$

where $\nu = 1, \ldots, |\nu|$ and R is a *coupling matrix* that is defined as follows:

$$R^{\nu,\alpha} = \begin{cases} 1 & \text{if } X_\nu \text{ and } Z_\alpha \text{ connected,} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Table 2 Update rules for different p_ν values

p_ν	Cost function	Multiplicative update rule
0	Euclidean	$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha}\phi_\nu^{-1}\Delta_{\alpha,\nu}(M_\nu \circ X_\nu)}{\sum_\nu R^{\nu,\alpha}\phi_\nu^{-1}\Delta_{\alpha,\nu}(M_\nu \circ \hat{X}_\nu)}$
1	Kullback–Leibler	$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha}\phi_\nu^{-1}\Delta_{\alpha,\nu}(M_\nu \circ \hat{X}_\nu^{-1} \circ X_\nu)}{\sum_\nu R^{\nu,\alpha}\phi_\nu^{-1}\Delta_{\alpha,\nu}(M_\nu)}$
2	Itakura–Saito	$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha}\phi_\nu^{-1}\Delta_{\alpha,\nu}(M_\nu \circ \hat{X}_\nu^{-2} \circ X_\nu)}{\sum_\nu R^{\nu,\alpha}\phi_\nu^{-1}\Delta_{\alpha,\nu}(M_\nu \circ \hat{X}_\nu^{-1})}$

Note that, distinct from PLTF model, there are multiple visible index sets ($V_{0,\nu}$) in the GCTF model, each specifying the attributes of the observed tensor X_ν.

The inference, i.e., estimation of the shared latent factors Z_α, can be achieved via iterative optimization (see Yilmaz et al. 2011). For non-negative data and factors, one can obtain the following compact fixed point equation where each Z_α is updated in an alternating fashion fixing the other factors $Z_{\alpha'}$ for $\alpha' \neq \alpha$

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha}\phi_\nu^{-1}\Delta_{\alpha,\nu}(M_\nu \circ \hat{X}_\nu^{-p_\nu} \circ X_\nu)}{\sum_\nu R^{\nu,\alpha}\phi_\nu^{-1}\Delta_{\alpha,\nu}(M_\nu \circ \hat{X}_\nu^{1-p_\nu})}, \tag{11}$$

where M_ν is a 0–1 mask array with $M_\nu(v_{0,\nu}) = 1\,(M_\nu(v_{0,\nu}) = 0)$ if $X_\nu(v_{0,\nu})$ is observed (missing). Here, p_ν determines the cost function as in (4) while dispersion parameter ϕ_ν is used for data driven regularization and weighting in coupled factorization of heterogeneous datasets. In Şimşekli et al. (2013b), they tackle learning the dispersion parameters ϕ_ν when $p \in \{0, 1, 2, 3\}$ by using a probabilistic approach, which makes use of the relation between the β-divergence and the family of Tweedie distributions and enables to find the dispersion parameters by maximizing the likelihood.

It is possible to choose different cost functions (different p_ν) for each observed data in a coupled model if each X_ν is modeled by a different type of distribution. Here, we solved update equations under the assumption of each observation tensor is modeled by the same type of distribution having the same dispersion parameter. This results in the same cost function (p_ν) for all the observed tensors X_ν and we can cancel out the dispersion parameters from the update equations.

See Table 2 for update rules for different p_ν values. In this iteration, the key quantity is the $\Delta_{\alpha,\nu}$ function that is defined as follows:

$$\Delta_{\alpha,\nu}(A) = \left[\sum_{v_{0,\nu} \cap \bar{v}_\alpha} A\left(v_{0,\nu}\right) \sum_{\bar{v}_0 \cap \bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z_{\alpha'}\left(v_{\alpha'}\right)^{R^{\nu,\alpha'}} \right]. \tag{12}$$

4 Link prediction with coupled tensor factorization

In this section, by using the GCTF framework, we address the missing link prediction task using different coupled models and loss functions on two real datasets, i.e., a small

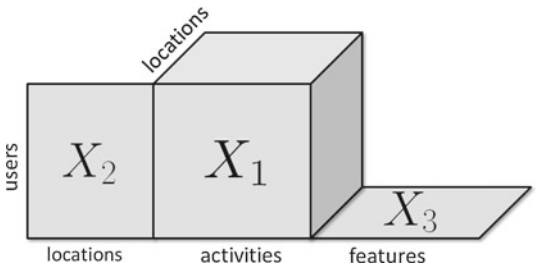


Fig. 1 UCLAF dataset represented in the form of a third-order tensor coupled with two matrices in two different modes

dataset called UCLAF[2] and a large-scale dataset called Digg.[3] We are not restricted to a specific model topology since the GCTF framework enables us to design application-specific models.

The choice of a particular factorization is strongly guided by the needs of an application, and there are some methods which are used to determine the right factorization model. First, the marginal likelihood of the observed data under a tensor factorization model is often necessary for certain problems such as model selection. This quantity can be estimated from variational Bayesian approach and the Gibbs output which is known as the Chib's method. Variational Bayes is applied to GCTF in Ermis and Cemgil (2013) and Chib's method is applied to PLTF in Simsekli and Cemgil (2012) in order to estimate the marginal likelihood for the tensor factorization frameworks. By computing the marginal likelihood, we can compare the tensor factorization models and choose the best model for a dataset. However, computing the marginal likelihood requires additional computational cost. Second one is to do cross-validation type experiments on each dataset and compare performances of the factorization models by omitting the known links from the dataset then making prediction for these links. Our simulations are close to the second method. At the beginning, we accept different percentages of links as missing, then predict the values of these links. Here, we first describe the datasets and then discuss the suitable factorization models by the defined method without computing the marginal likelihood in order to save time.

4.1 UCLAF dataset

UCLAF dataset (Zheng et al. 2010) is extracted from the GPS data that include information of three types of entities: user, location and activity (see Fig. 1 for an illustration of the data). The relations between user–location–activity triplets are used to construct a three-way tensor X_1. In tensor X_1, an entry $X_1(i,\ j,\ k)$ indicates the frequency of user i visiting location j and doing activity k there; otherwise, it is 0. Since we address the link prediction problem in this study, we define the user–location–activity tensor X_1 as:

$$X_1(i,\ j,\ k) = \begin{cases} 1 & \text{if user } i \text{ visits location } j \text{ and performs activity } k \text{ there,} \\ 0 & \text{otherwise.} \end{cases}$$

[2] http://www.cse.ust.hk/~vincentz/aaai10.uclaf.data.mat.

[3] http://www.public.esu.edu/~ylin56/kdd09sup.html.

To construct the dataset, raw GPS points were clustered into 168 meaningful locations and the user comments attached to the GPS data were manually parsed into activity annotations for the 168 locations. Consequently, the data consists of 164 users, 168 locations and 5 different types of activities, i.e., 'Food and Drink', 'Shopping', 'Movies and Shows', 'Sports and Exercise', and 'Tourism and Amusement' (Zheng et al. 2010).

The collected data also includes additional side information: the user–location preferences from the GPS trajectory data and the location features from the POI (points of interest) database, represented as the matrix X_2 and X_3, respectively. In our model the user–location preferences matrix has entries $X_2(i, m)$ of size $I \times J$, where I is the number of users and J is the number of locations. However, in our model we use a separate index m for the location index in X_2 instead of j. The rationale behind this choice is to relax the model as the entries in X_1 and X_2 are measuring distinct quantities: $X_2(i, m)$ represents the frequency of user i visiting location m and stayed there over a time threshold while X_1 only indicates an activity by a specific user i at location j. The relation between the location entries j and m in X_1 and X_2 are coupled via a common factor over the users. Finally, we represent the location–feature values with matrix X_3 of size $J \times N$, where J is the number of locations, that has the same location type in X_1, and N is the number of features. In particular, an entry $X_3(j, n)$ represents the number of different POIs at a location j. Using the location features, we could gain information about location similarities.

In this dataset, 18 users have no location and activity information. Therefore, we have used the data from the remaining 146 users. In order to decrease the effect of outliers, location–feature matrix is preprocessed as follows: $X_3(j, n) = 1 + \log(X_3(j, n))$ if $X_3(j, n) > 0$; otherwise, $X_3(j, n) = 0$. In our experiments, number of users is $I = 146$, number of locations $J = 168$, number of activities $K = 5$ and number of location features $N = 14$.

We have a three-way observation tensor X_1 with elements 0 and 1, where 0 denotes a known absent link and 1 denotes a known present link, and two auxiliary matrices X_2 and X_3 that provide side information. Our aim is to restore the missing links in X_1. This is a difficult link prediction problem since X_1 contains less than 1 % of all possible links or an entire slice of X_1 may be missing. Using low-rank factorization of a tensor to estimate missing entries will be ineffective, in particular, in the case of structured missing data such as missing slices.

In order to fill in the missing links in tensor X_1, we form four different coupled models changing in the way tensor X_1 is factorized, i.e., using a CP, Tucker, Paratuck-style (Harshman et al. 1996) or some arbitrary factorization. For all models, we use KL divergence and EUC as cost functions in our non-negative decomposition problems. Table 3 summarizes the models and the corresponding equations.

Table 3 Different coupled models on UCLAF dataset

Models	Equation numbers
Model 1 (CP)	13–15
Model 2 (Tucker)	17–19
Model 3 (Paratuck)	20–22
Model 4 (Arbitrary)	23–25

4.1.1 Model 1 (CP)

In the first model, we applied the coupled approach to a CP-style tensor factorization model by analyzing the tensor X_1 jointly with the additional matrices X_2 and X_3 in order to solve the sparsity problem in X_1 effectively. This gives us the following model:

$$\hat{X}_1(i,\ j,\ k) = \sum_r A(i,\ r)B(j,\ r)C(k,\ r), \tag{13}$$

$$\hat{X}_2(i,\ m) = \sum_r A(i,\ r)D(m,\ r), \tag{14}$$

$$\hat{X}_3(j,\ n) = \sum_r B(j,\ r)E(n,\ r). \tag{15}$$

Here, we have three observed tensors, that share common factors; therefore, we have a coupled tensor factorization problem. The coupling matrix R with $|\alpha| = 5,\ |\nu| = 3$ for this model is defined as follows:

$$R = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \text{with} \quad \begin{aligned} \hat{X}_1 &= \textstyle\sum A^1 B^1 C^1 D^0 E^0, \\ \hat{X}_2 &= \textstyle\sum A^1 B^0 C^0 D^1 E^0, \\ \hat{X}_3 &= \textstyle\sum A^0 B^1 C^0 D^0 E^1. \end{aligned} \tag{16}$$

Note that, X_1 and X_2 share the common factor matrix A with entries $A(i,\ r)$; we can interpret each row of $A(i,\ :)$ as user i's latent position in a $|r|$ dimensional 'preferences' space. The factor matrix B with entries $B(j,\ r)$ represents the latent position of the location j in the same preferences space. The user i at location j tends to make the activity k where the weight $A(i,\ r)B(j,\ r)$ is large for at least one r, i.e., there is a match between the users preference and what the location 'has to offer'. The location specific factor B is also influenced by the location–feature matrix X_3.

We show the computation for A, i.e. for Z_1, which is the common factor of X_1 and X_2 and the computation for B, i.e. for Z_2, which is the common factor of X_1 and X_3 in Appendix 6.

4.1.2 Model 2 (Tucker)

Following the same line of thought, we apply the coupled approach using a Tucker factorization to form our second model, which is as follows:

$$\hat{X}_1(i,\ j,\ k) = \sum_{p,q,r} A(i,\ p)B(j,\ q)C(k,\ r)D(p,\ q,\ r), \tag{17}$$

$$\hat{X}_2(i,\ m) = \sum_p A(i,\ p)E(m,\ p), \tag{18}$$

$$\hat{X}_3(j,\ n) = \sum_r B(j,\ q)F(n,\ q). \tag{19}$$

In this model, once again, the factor A is shared by X_1 and X_2, while the factor B is shared by X_1 and X_3. In contrast to the coupled CP model in (13), this model assumes that user i at location j tends to make the activity k with the weight $\sum_{p,q} A(i,\ p)B(j,\ q)C(k,\ r)D(p,\ q,\ r)$. Here, a latent preference space interpretation is less intuitive but the model has more freedom to represent the dependence.

4.1.3 Model 3

In this model, we apply the coupled approach to a Paratuck-style (Harshman et al. 1996) tensor model by analyzing the tensor X_1 jointly with the additional matrices X_2 and X_3. This gives us the following model:

$$\hat{X}_1(i,\ j,\ k) = \sum_{p,q} A(i,\ p)B(j,\ q)C(k,\ p)D(k,\ q)G(p,\ q), \tag{20}$$

$$\hat{X}_2(i,\ m) = \sum_{p} A(i,\ p)E(m,\ p), \tag{21}$$

$$\hat{X}_3(j,\ n) = \sum_{q} B(j,\ q)F(n,\ q). \tag{22}$$

4.1.4 Model 4

As our final model, we use an arbitrary tensor model to jointly analyze tensor X_1 with the additional matrices X_2 and X_3. Here, we introduce a new dummy index d and call this model *Model 4*, which is defined as follows:

$$\hat{X}_1(i,\ j,\ k) = \sum_{d,r} A(i,\ d)B(d,\ r)C(j,\ r)D(k,\ r), \tag{23}$$

$$\hat{X}_2(i,\ m) = \sum_{d,r} A(i,\ d)B(d,\ r)E(m,\ r), \tag{24}$$

$$\hat{X}_3(j,\ n) = \sum_{r} C(j,\ r)F(n,\ r). \tag{25}$$

4.2 Digg dataset

We address link prediction problem also on a large-scale dataset collected from Digg in order to show the scalability of the proposed approach. Digg is a social news resource that allows users to submit, Digg and comment on news stories. Lin et al. (2009) have collected data from a large set of user actions from Digg. The dataset is a subset of data scrapped from Digg by Choudhury et al. (2009) during January 2009. It includes stories, users and their actions (submit, Digg, comment and reply) with respect to the stories, as well as the explicit friendship (contact) relation among these users. It also includes the topics of the stories and keywords extracted from the titles of stories. There are five types of entities: user, story, comment, keyword and topic and six relationships among them (see Lin et al. 2009 for a comprehensive illustration of relations).

Table 4 Different coupled models on Digg dataset

Models	Equation numbers		
	Comment prediction (with X_1 and X_2)	Comment prediction (with X_1–X_3)	Digg prediction
Model 1 (CP)	26, 27	31–33	37, 38
Model 2 (Tucker)	29, 30	34–36	39, 40

We will use three relationships in this study: user–story–comment (R1), story–keyword–topic (R2) and user–story (R3). Lin et al. (2009) extract tuples with timestamps ranging from 1 August to 27 August 2008, segment the data duration into nine time slots (i.e. every 3 days), and construct a sequence of data tensors for each dynamic relation in order to study the data evolution. Except for the contact relation, all relations in this dataset have timestamps. However, in our work, since we are not modeling the evolution in time, we integrate the nine segments together and evaluate missing link prediction tasks on this integrated data. The total number of tuples in each integrated data tensor per relation is 151779, 1157529 and 94551, respectively. The prediction results are compared with the actual diggs and comments as ground truth.

Based on the Digg scenario, we design two prediction tasks on Digg dataset:

(i) comment prediction: what stories a user will comment on,
(ii) Digg prediction: what stories a user will Digg.

For comment and Digg prediction, we form different coupled models. Table 4 summarizes these models and the corresponding equations.

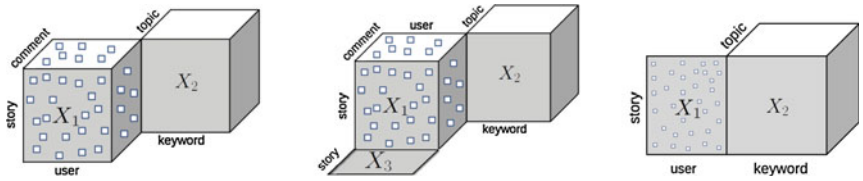
4.2.1 Comment prediction

For comment prediction, the relation between the user–story–comment is used to construct tensor X_1 of size $I \times J \times K$ where the number of users is $I = 9{,}583$, the number of stories is $J = 44{,}005$ and the number of comments is $K = 241{,}800$. X_1 is defined as:

$$X_1(i,\ j,\ k) = \begin{cases} 1 & \text{if user } i \text{ comments on story } j \text{ with comment } k, \\ 0 & \text{otherwise.} \end{cases}$$

Additionally, the data includes the topics of the stories and extracted keywords from the stories' titles. We represent this data as the three-way tensor X_2. In our model the story–keyword–topic tensor has entries $X_2(j,\ m,\ n)$ of size $J \times M \times N$, where the number of stories is $J = 44{,}005$, the number of keywords is $M = 13{,}714$ and the number of topics is $N = 51$.

Our aim is to restore the missing links in X_1 (see Fig. 2a for an illustration of the modeled data). Here, X_1 contains less than 0.07 % of all possible links. We form two coupled models in order to fill in the missing links in tensor X_1 through joint analysis of X_1 and X_2. For both models, we use EUC distance, KL divergence and IS



(a) Coupled datasets used for comment prediction.

(b) Coupled datasets used for comment prediction with different relational contexts.

(c) Coupled datasets used for digg prediction.

Fig. 2 Comment and Digg prediction on Digg dataset

divergence. We also explore the behaviour of the models using various cost functions, i.e., $p \in [0, 2]$, based on β-divergences.

Model 1 (*C*P) in the first model, we applied the coupled approach to a CP-style tensor factorization model by analyzing the tensor X_1 jointly with the additional tensor X_2 as follows:

$$\hat{X}_1(i, j, k) = \sum_r A(i, r)B(j, r)C(k, r), \tag{26}$$

$$\hat{X}_2(j, m, n) = \sum_r B(j, r)D(m, r)E(n, r). \tag{27}$$

Here, we have two observed tensors, X_1 and X_2, that share factor matrix B. The coupling matrix R with $|\alpha| = 5$, $|\nu| = 2$ for this model is defined as follows:

$$R = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad \text{with} \quad \begin{matrix} \hat{X}_1 = \sum A^1 B^1 C^1 D^0 E^0, \\ \hat{X}_2 = \sum A^0 B^1 C^0 D^1 E^1. \end{matrix} \tag{28}$$

We can interpret each row of $B(j, :)$ as story j's latent position in a $|r|$ dimensional *preferences* space. The factor matrix A with entries $A(i, r)$ represents the latent position of user i in the same preferences space. The user i tends to comment on the story j with comment k where the weight $A(i, r)B(j, r)C(k,)$ is large for at least one r.

Model 2 (*T*ucker) we also apply the coupled approach using a Tucker factorization as follows:

$$\hat{X}_1(i, j, k) = \sum_{p,q,r} A(i, p)B(j, q)C(k, r)D(p, q, r), \tag{29}$$

$$\hat{X}_2(j, m, n) = \sum_q B(j, q)E(m, q)F(n, q), \tag{30}$$

where factor B is shared by X_1 and X_2. In contrast to the coupled CP model sketched in Eq. 26, this model assumes that user i tends to comment on the story j with comment k, with the weight $\sum_{p,q} A(i, p)B(j, q)C(k, r)D(p, q, r)$.

Comment Prediction with different relational contexts we observe that different combinations of relations affect the prediction performance. In addition to the relation between user–story–comment triplets (represented by tensor X_1) and the relation

between story–keyword–topic triplets (represented by X_2), here we also incorporate the relation between users and stories represented by matrix X_3 (Fig. 2b).

In our model the user–story–comment tensor has entries $X_1(i,\ j,\ k)$. However, we use a separate index t for the story index in X_3 instead of j. The rationale behind this choice is to relax the model as the entries in X_1 and X_3 are measuring distinct quantities: $X_1(i,\ j,\ k)$ represents whether the user i comments on story j with comment k, while X_3 only indicates a vote (i.e. Digg) by a specific user i on story t. The relation between story entries j and t in X_1 and X_3 are coupled via a common factor over the users.

We form two coupled models for comment prediction through joint factorization of $X_1,\ X_2$ and X_3 in order to fill in the missing links in tensor X_1. For both models, we use EUC distance, KL divergence and IS divergence as cost functions.

Model 1 (*C*P) in the first model, we again applied the coupled approach to a CP-style tensor model by analyzing tensor X_1 jointly with additional tensors X_2 and X_3 as follows:

$$\hat{X}_1(i,\ j,\ k) = \sum_r A(i,\ r)B(j,\ r)C(k,\ r), \tag{31}$$

$$\hat{X}_2(j,\ m,\ n) = \sum_r B(j,\ r)D(m,\ r)E(n,\ r), \tag{32}$$

$$\hat{X}_3(i,\ t) = \sum_r A(i,\ r)F(t,\ r). \tag{33}$$

Here, we have three observed tensors with common factors. Note that X_1 and X_3 share factor matrix A with entries $A(i,\ r)$; we can interpret each row of $A(i,\ :)$ as user i's latent position in a $|r|$ dimensional *preferences* space.

Model 2 (*T*ucker) likewise, we applied a Tucker-based coupled approach as follows:

$$\hat{X}_1(i,\ j,\ k) = \sum_{p,q,r} A(i,\ p)B(j,\ q)C(k,\ r)G(p,\ q,\ r), \tag{34}$$

$$\hat{X}_2(j,\ m,\ n) = \sum_q B(j,\ q)D(m,\ q)E(n,\ q), \tag{35}$$

$$\hat{X}_3(i,\ t) = \sum_p A(i,\ p)F(t,\ p). \tag{36}$$

4.2.2 Digg prediction

For Digg prediction, the relation between users and stories is used to construct matrix X_1 of size $I \times J$ where the number of users is I=9,583 and the number of stories is J=44,005. The user–story matrix X_1 is defined as:

$$X_1(i,\ j) = \begin{cases} 1 & \text{if user } i \text{ votes (i.e. Digg) on news stories } j, \\ 0 & \text{otherwise.} \end{cases}$$

Additionally, the data includes the topics of the stories and extracted keywords from titles of stories. We represent this data as a three-way tensor X_2. In our model the story–keyword–topic tensor has entries $X_2(j, k, m)$ of size $J \times K \times M$, where the number of stories $J = 44{,}005$, the number of keywords is $K = 13{,}714$ and the number of topics is $M = 51$.

Here, our aim is to restore the missing links in X_1 (Fig. 2c). This is also a difficult link prediction problem since X_1 contains less than 0.008 % of all possible links. Once again, we form coupled models based on CP and Tucker models in order to fill in the missing links in matrix X_1. For both models, as cost functions, we use EUC distance, KL divergence, IS divergence as well as various cost functions, i.e., $p \in [0, 2]$, based on β-divergences.

Model 1 (*C*P) we applied the coupled approach based on a CP-style tensor model by analyzing matrix X_1 jointly with tensor X_2 as follows:

$$\hat{X}_1(i, j) = \sum_r A(i, r)B(j, r), \tag{37}$$

$$\hat{X}_2(j, k, m) = \sum_r B(j, r)C(k, r)D(m, r). \tag{38}$$

Here, X_1 and X_2 share factor matrix B with entries $B(j, r)$; we can interpret each row of $B(j, :)$ as story j's latent position in a $|r|$ dimensional *preferences* space. The factor matrix A with entries $A(i, r)$ represents the latent position of user i in the same preferences space. The user i tends to vote for the story j, where the weight $A(i, r)B(j, r)$ is large for at least one r, i.e., there is a match between the users preference and what the story 'has to offer'.

Model 2 (*T*ucker) we also use a Tucker model for the coupled approach as follows:

$$\hat{X}_1(i, j) = \sum_p A(i, p)B(j, p), \tag{39}$$

$$\hat{X}_2(j, k, m) = \sum_{p,q,r} B(j, p)C(k, q)D(m, r)G(p, q, r). \tag{40}$$

5 Experimental results

This section reports our experimental study on two real world datasets: UCLAF and Digg. For both datasets, we first demonstrate that coupled tensor factorizations outperform low-rank approximations of a single tensor in terms of missing link prediction. Then, within the context of coupled tensor factorizations, we compare different tensor models and loss functions including the ones previously proposed in the literature (see Table 1) and show that selection of the tensor model and loss function is significant in terms of link prediction performance, especially when the data is sparse. Our experiments demonstrate that loss functions that have not been studied for link prediction before, such as IS-divergence, outperform the commonly-used loss functions.

Furthermore, we study the case with completely missing slices, which corresponds to the cold-start problem in our link prediction setting and demonstrate that it is still

possible to predict missing links using the proposed coupled models whereas low-rank approximations of a single tensor would fail to do so.

5.1 Computational environment

All experiments were performed using MATLAB 2010b on 2.4 GHz Core i5 520M processor and 4 GB RAM. Timings were performed using MATLABs tic and toc functions.

5.2 Stopping conditions

We use the relative change in error value as a stopping condition. The error at iteration i is calculated as $e^{(i)} = \frac{1}{2}\|X^{(i)} - \hat{X}^{(i)}\|^2$ and the algorithm stops when $|e^{(i)} - e^{(i-1)}|/e^{(i-1)} \leq 10^{-6}$ where i is the iteration number. In addition, the maximum number of iterations is set to 1,000. We observe that the algorithm has generally stopped due to the relative change criterion.

5.3 Computational complexity

Assuming that all datasets have equal number of dimensions, i.e., a tensor is an $N \times N \times N$ array while the coupled matrix is of size $N \times N$, then the leading term in the computational complexity of the coupled model will be due to the updates for the tensor model. For an R-component CP model, for instance, that would be $O(N^3R)$.

If a large number of entries is missing, then mask tensor M is sparse. In this case, there is no need to allocate storage for every entry of the tensor X. Instead, we can store and work with just the known values, making the method efficient in both storage and time. Our approach also has ability to perform sparse computations, enabling it to scale to very large real datasets using specialized sparse data structures, significantly reducing the storage and computation costs. When we take into account the sparsity pattern of the data, the time complexity of each iteration is roughly $O(N)$, which is linear in terms of the total number of non-missing entries N. We also give empirical results in Appendix 6.

5.4 Evaluation metrics

In our experiments, as evaluation metrics, we use area under the receiver operating characteristic (ROC) curve (AUC) and P@K (the precision of the top K results) for link prediction results and root mean square error (RMSE) for tensor completion results.

5.5 RMSE

RMSE is a measure of the 'average' error, weighted according to the square of the error. In our experiments, we use RMSE to measure the tensor reconstruction performance.

5.6 AUC

Link prediction datasets are characterized by extreme imbalance, i.e., the number of links known to be present is often significantly less than the number of edges known to be absent. This issue motivates the use of AUC as a performance measure since AUC is viewed as a robust measure in the presence of imbalance (Stäger et al. 2006).

5.7 P@K

Precision at k (P@K) measures the precision at a fixed number of retrieved items (i.e., top K) of the ordered list r' and the unordered list r (Sanderson 2010). Assume $TopK$ and $TopK'$ are the retrieved items of r and r', respectively, then the P@K is defined as $P@K = \frac{|TopK \cap TopK'|}{K}$.

We use P@K to measure the performance of prediction. As might be expected, the accuracy of link prediction also varies according to the precision measure chosen. Due to its robustness P@K is a frequently used measure in the domain of information retrieval and machine learning (Spiegel et al. 2011). We compute the precision based on the top 10 stories retrieved for each user on Digg dataset. The overall P@10 for the set of users is computed by taking the mean of P@10 per user.

The following results show the *average link prediction performance of 10 independent runs* in terms of AUC, ROC curve and P@K.

5.8 UCLAF dataset

In this section, we assess the performance of the coupled models proposed in Sect. 4.1 in terms of tensor completion and/or missing link prediction.

5.8.1 Experimental setting

We design experiments to evaluate the performance of our models in terms of link prediction. By setting different amounts of data to missing in user–location–activity tensor X_1, we compare the following models using both KL-divergence and the EUC as cost functions:

- *Low-rank approximations of a single tensor* (i) CP and (ii) Tucker factorization of user–location–activity tensor X_1,
- *Coupled tensor factorizations* (i) CP factorization of X_1 coupled with factorization of user–location matrix X_2 and location–feature matrix X_3 (Eqs. 13–15), (ii) Tucker factorization of X_1 coupled with factorization of X_2 and X_3 (Eqs. 17–19), (iii) Model 3 (Eqs. 20–22), and (iv) Model 4 (Eqs. 23–25).

We use two patterns of missing data: (i) randomly missing entries and (ii) randomly missing slices. In all experiments, *number of components*, i.e., number of columns in each factor matrix, Z_i, is set to 2.

Table 5 RMSE for different models with different percentages of training data

Models	EUC		KL	
	30 %	50 %	30 %	50 %
CP	0.27 ∓ 0.03	0.28 ∓ 0.04	0.24 ∓ 0.03	0.23 ∓ 0.03
Tucker	0.26 ∓ 0.02	0.26 ∓ 0.04	0.22 ∓ 0.02	0.22 ∓ 0.02
Coupled (CP)	0.24 ∓ 0.01	0.23 ∓ 0.02	0.19 ∓ 0.02	0.18 ∓ 0.02
Coupled (Tucker)	0.22 ∓ 0.01	0.22 ∓ 0.02	0.18 ∓ 0.01	0.18 ∓ 0.01
PCLAF (Zheng et al. 2010)	0.30 ∓ 0.01	0.29 ∓ 0.01	–	–

5.8.2 Results

Tensor completion Table 5 shows tensor completion performances of standard CP and Tucker models, coupled models and PCLAF (Zheng et al. 2010). PCLAF is a personalized collaborative location and activity filtering algorithm, which uses a collective tensor and matrix factorization. In addition to the data that we have used in our models, PCLAF uses user–user and activity–activity similarity matrices in UCLAF dataset. Also, PCLAF uses CP tensor factorization model and EUC distance as cost function. For PCLAF algorithm, they run the experiments five times, and report the average RMSE scores. Specifically, at each trial, they randomly split some percentage (30 and 50 %) of the existing tensor entries for training and hold out the other for testing. We also set the same amount of missing entities randomly and report the average RMSE scores of 10 independent runs. Hence, our results are comparable to PCLAF algorithm's results. Eventually, we observe that our models outperform the PCLAF approach, which has outperformed many collaborative filtering methods in Zheng et al. (2012), especially when we use KL divergence which is a lot more natural than a EUC cost for this data.

Link prediction in order to demonstrate the power of coupled analysis, we compared the link prediction performance of standard CP and Tucker models with coupled ones using EUC and KL cost functions at different amounts, i.e., {40, 60, 80, 90, 95}, of randomly unobserved elements. For all cases, coupled models outperform the standard models clearly. Figure 3 shows the comparison of CP and coupled CP models with different cost functions when 80 % of the data is missing. As we can see, the coupled models that try to use as much additional information as possible to help alleviate the data sparsity issue perform better than the standard models; in particular, when the percentage of missing data is high (see Table 6). When the fraction of missing data was more than 80 %, the standard models could not find a solution.

In order to demonstrate the effect of the cost function modeling the data, we have also carried out experiments on both coupled CP and Tucker models at different missing data fractions. For all cases, the KL cost function seems to perform better than EUC, especially when the fraction of missing entries is high. Figure 4 illustrates the performance of EUC distance and KL divergence for both coupled CP and Tucker models when 90 % of the data is unobserved.

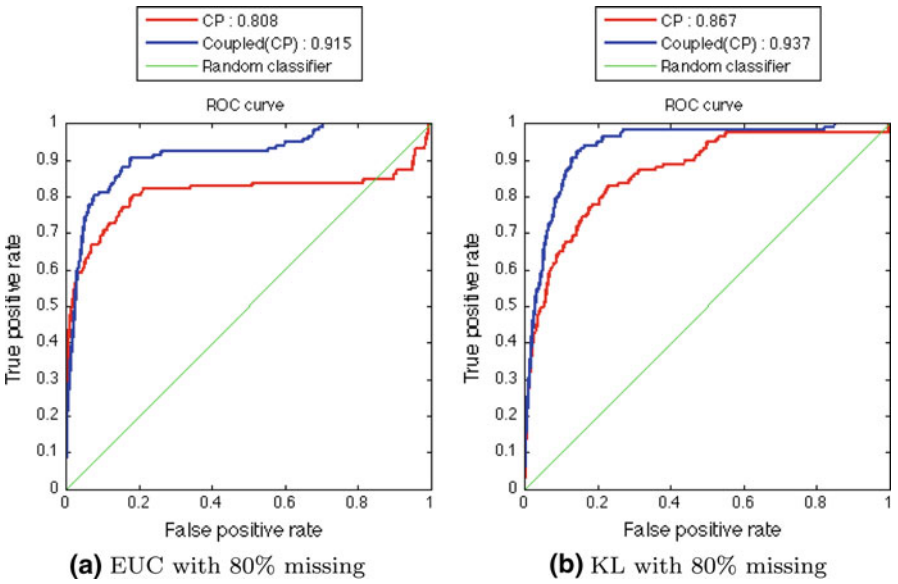


Fig. 3 Comparison of CP and coupled (CP) models

Table 6 Link prediction results on UCLAF with different experimental settings

	40 %		80 %		90 %	
	EUC	KL	EUC	KL	EUC	KL
CP	0.920	0.946	0.808	0.867	–	–
Tucker	0.943	0.960	0.896	0.917	–	–
CP (coupled)	0.951	0.968	0.915	0.937	0.813	0.869
Tucker (coupled)	0.965	0.983	0.934	0.948	0.871	0.908

Figure 5 shows the comparison of coupled CP and Tucker models in order to illustrate the tensor model which models the data best. We can see that Tucker model outperforms the CP model; because Tucker model is more flexible due to the full core tensor which is helpful for us to explore the structural information embedded in the data. In Fig. 6, we also compare coupled CP and coupled Tucker with some arbitrary factorizations, i.e., Model 3 (given in Eqs. 20–22) and Model 4 (given in Eqs. 23–25). We can see that Tucker model outperforms all the other models.

Finally, we demonstrate the effect of cardinality of latent indices R on link prediction performance. Figure 7 illustrates the performance of coupled CP model when $R=2$ and 5 for both EUC and KL divergences when 90 % of the data is unobserved. It is clear that the average scores for both values of R are quite close. We use $R=2$ for the rest of the experiments.

Missing slice we also study the *cold-start problem*, which is particularly important in link prediction because we may often have new users starting to use an application, e.g., a location–activity recommender system. Since they are new users, they will

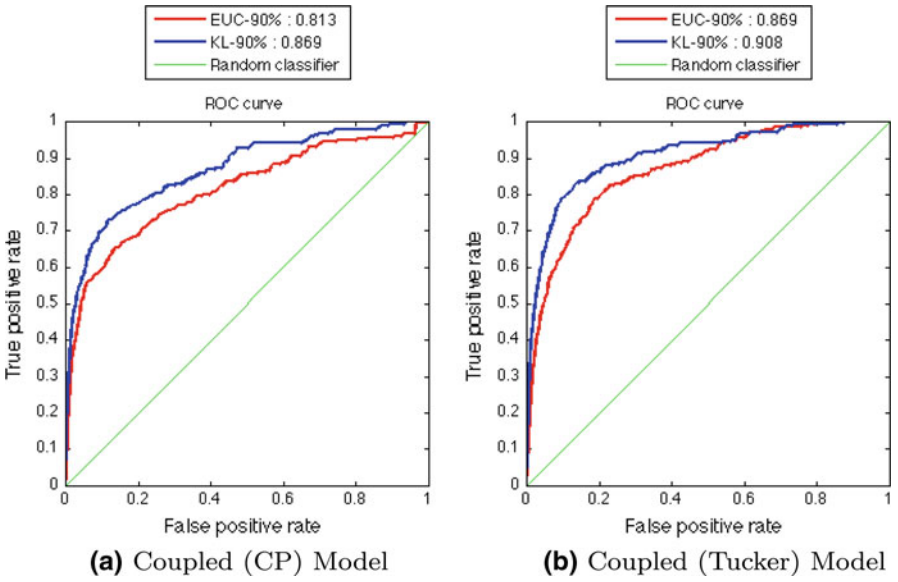


Fig. 4 Comparison of EUC distance and KL divergence with 90 % missing data

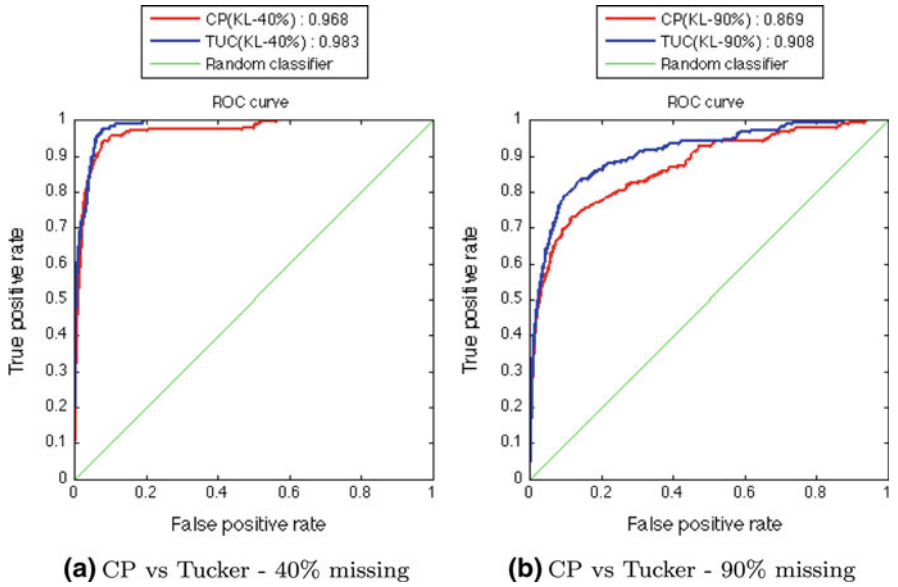
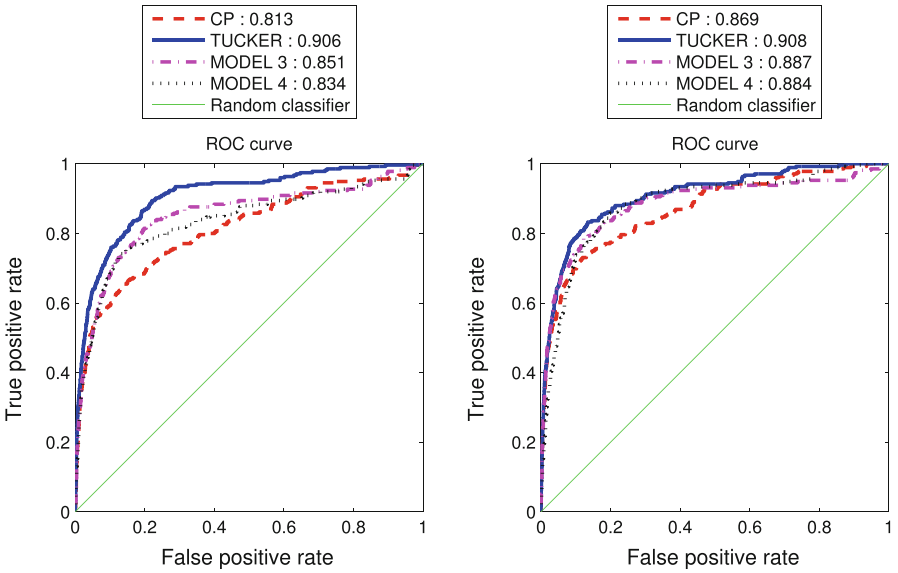


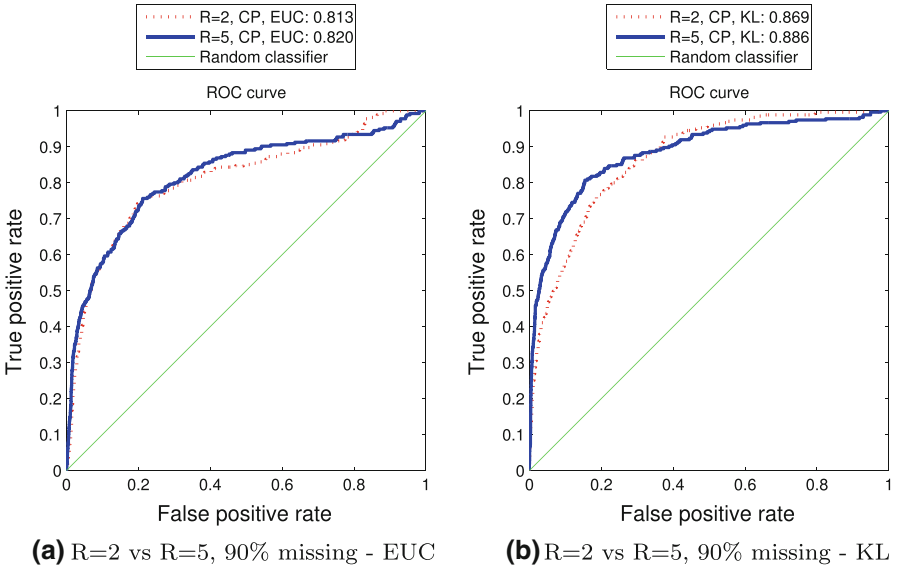
Fig. 5 Comparison of coupled CP and Tucker models with KL

have no entry in X_1, i.e., a completely missing slice (see Fig. 8 for illustration of the problem). It is not possible to reconstruct a missing slice of a tensor using its low-rank approximation. A similar argument is valid in the case of matrices for completely missing rows/columns (Candès and Plan 2010). In such cases, additional sources of



(a) Arbitrary models - EUC **(b)** Arbitrary models - KL

Fig. 6 Comparison of four different coupled tensor factorization models with 90 % missing data

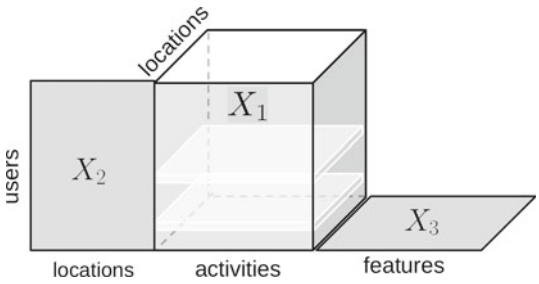


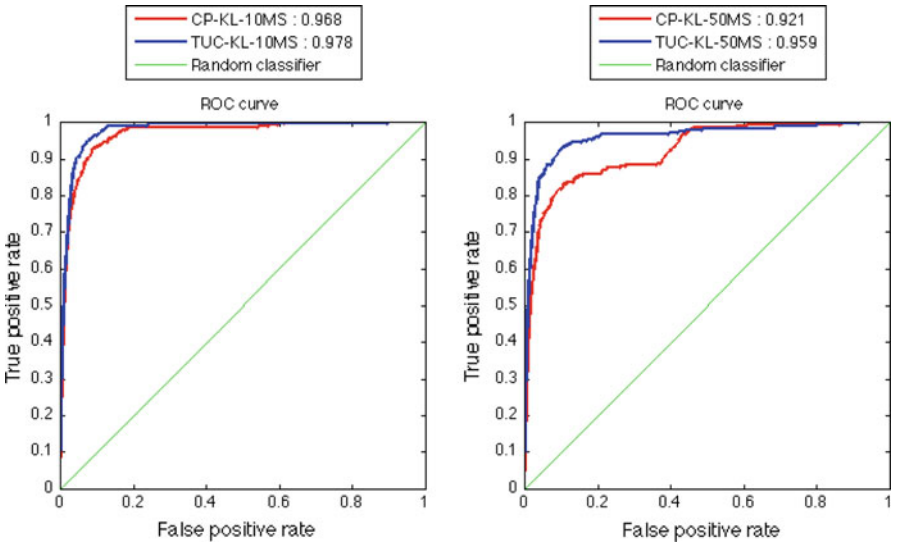
(a) R=2 vs R=5, 90% missing - EUC **(b)** R=2 vs R=5, 90% missing - KL

Fig. 7 Comparison of $R=2$ and 5 with CP model

information will be useful (Narita et al. 2011) to make recommendations to new users. We observe that our coupled models could predict the links when there is no information about a user in tensor X_1, by utilizing the additional sources of information. We test this case by setting randomly missing slices in X_1.

Fig. 8 Cold-start problem





(a) CP and Tucker - 10MS **(b)** CP and Tucker - 50MS

Fig. 9 Link prediction result with missing slices and KL cost

Figure 9 demonstrates the performance of coupled models with KL divergence when 10- and 50 users' data are missing. Also note that Tucker is superior to CP as the amount of missing data increases.

Table 6 summarizes the experimental results given in this section on UCLAF dataset in terms of AUC metric.

5.9 Digg dataset

In this section, we assess the performance of the coupled models proposed in Sect. 4.2 in terms of missing link prediction.

5.9.1 Experimental setting

We design experiments to evaluate the performance of our models given in Sect. 4.2 in terms of missing link prediction on Digg dataset. Based on the Digg scenario, we have

two prediction tasks on Digg dataset: (i) comment prediction and (ii) Digg prediction that are explained in Sect. 4.2.

For the first prediction task, by setting different amounts of data to missing in user–story–comment tensor $\mathcal{X}_1$, we compare the following models using EUC distance, KL and IS divergences as cost functions. We also assess the performance of various cost functions based on β-divergences.

- *Low-rank approximations of a single tensor* (i) CP and (ii) Tucker factorization of user–story–comment tensor $\mathcal{X}_1$,
- *Coupled tensor factorizations* (i) CP factorization of $\mathcal{X}_1$ coupled with factorization of story–keyword–topic tensor $\mathcal{X}_2$ (Eqs. 26, 27) and (ii) Tucker factorization of $\mathcal{X}_1$ coupled with factorization of $\mathcal{X}_2$ (Eqs. 29, 30),
- *Coupled tensor factorizations* (i) CP factorization of $\mathcal{X}_1$ coupled with factorization of $\mathcal{X}_2$ and user–story matrix $\mathbf{X}_3$ (Eqs. 31–33) and (ii) Tucker factorization of $\mathcal{X}_1$ coupled with factorization of $\mathcal{X}_2$ and $\mathbf{X}_3$ (Eqs. 34–36).

For the second prediction task, by setting different amounts of data to missing in user–story matrix $\mathbf{X}_1$, we compare the following models using EUC distance, KL and IS divergences as cost functions. Just like in comment prediction, in addition to these cost functions, we also consider additional loss functions based on β-divergences.

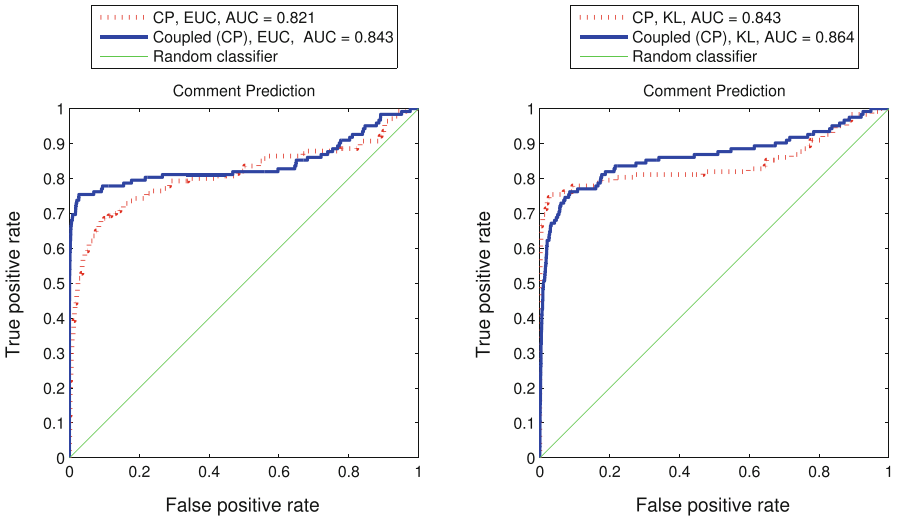
- *Coupled tensor factorizations* (i) matrix factorization of $\mathbf{X}_1$ coupled with CP factorization of story–keyword–topic tensor $\mathcal{X}_2$ (Eqs. 37, 38) and (ii) matrix factorization of $\mathbf{X}_1$ coupled with Tucker factorization of $\mathcal{X}_2$ (Eqs. 39, 40).

In all experiments, we set *number of components*, i.e., number of columns in each factor matrix, $\mathbf{Z}_i$, to 5. We consider both randomly missing entries and randomly missing slices.

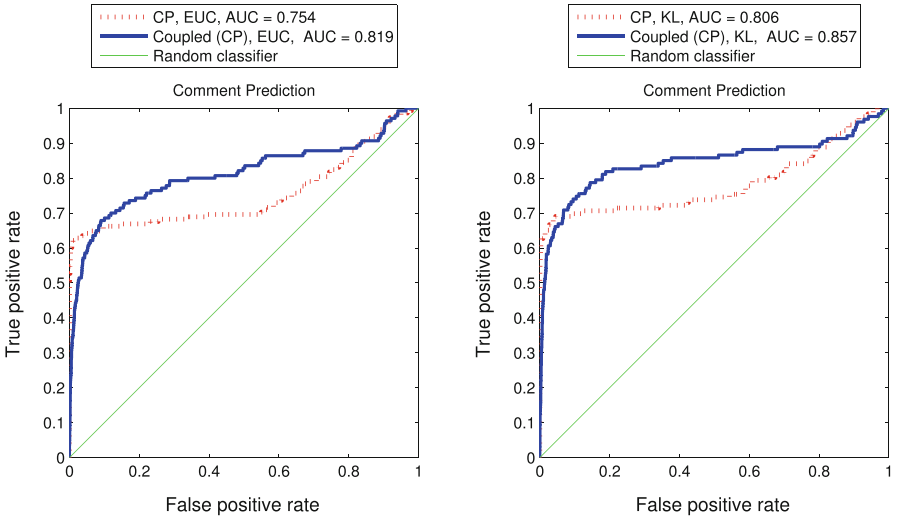
5.9.2 Results

In order to demonstrate the power of coupled analysis, we compared the link prediction performance of standard CP and Tucker models with coupled ones using EUC, KL and IS cost functions at different amounts, i.e., {40, 80, 90}, of randomly unobserved elements. Here, we show results of the experiments on *both comment and Digg prediction tasks*. For all cases, coupled models outperform the standard models clearly. Figure 10 shows the comparison of CP and coupled CP models with different cost functions when 40 and 80 % of the data are missing. As we can see, coupled models perform better than the standard models; in particular, when the percentage of missing data is high. When the fraction of missing data was more than 80 %, the standard models could not find a solution. In Fig. 10, we denote coupled models as coupled (CP) in the legend; in the rest of the figures, we use only CP or Tucker in the legends indicating coupled models.

In order to demonstrate the effect of the cost function modeling the data, we have also carried out experiments on both coupled CP and Tucker models at different missing data fractions. For all cases, the IS cost function seems to perform better than EUC and KL for both prediction tasks, especially when the fraction of missing entries is high. Figures 11 and 12 illustrate the performance of EUC distance, KL divergence



(a) EUC with 40% missing **(b)** KL with 40% missing



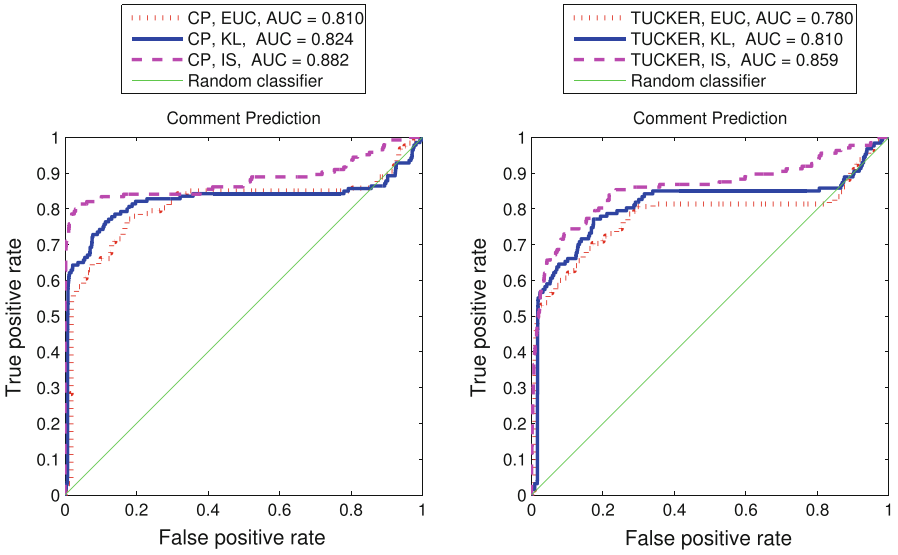
(c) EUC with 80% missing **(d)** KL with 80% missing

Fig. 10 Comparison of CP and coupled (CP) models for comment prediction

and IS divergence for both coupled CP and Tucker models when 40 and 90 % of the data is unobserved, for comment prediction and Digg prediction, respectively.

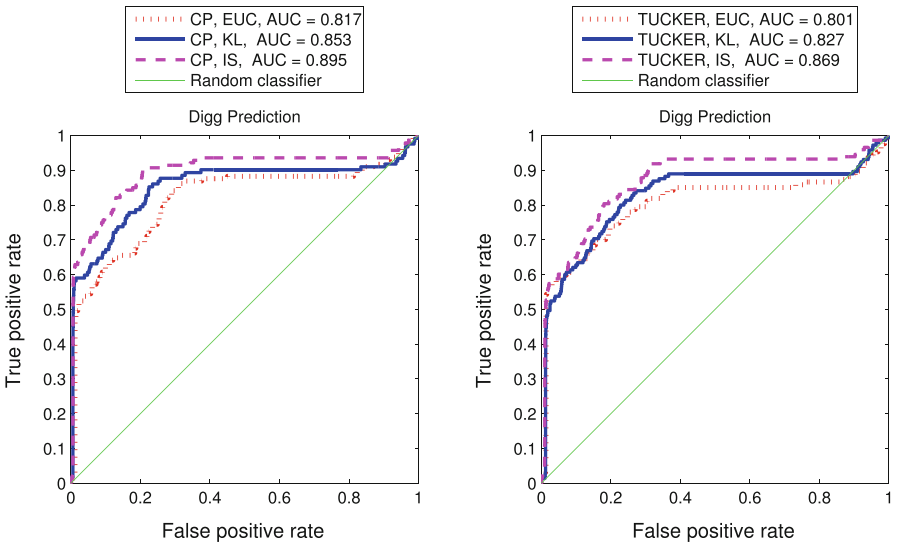
When the missing data rate becomes higher, the difference between performances of cost functions become clearer.

In Figs. 11 and 12, we also observe that CP model outperforms the Tucker model in terms of capturing the structural information embedded in the data. In addition,



(a) CP - 90% missing **(b)** Tucker - 90% missing

Fig. 11 Comparison of EUC, KL and IS on comment prediction of the models in Eqs. 26, 27 for CP and Eqs. 29, 30 for Tucker



(a) CP - 90% missing **(b)** Tucker - 90% missing

Fig. 12 Comparison of EUC, KL and IS on Digg prediction of the models in Eqs. 37, 38 for CP and Eqs. 39, 40 for Tucker

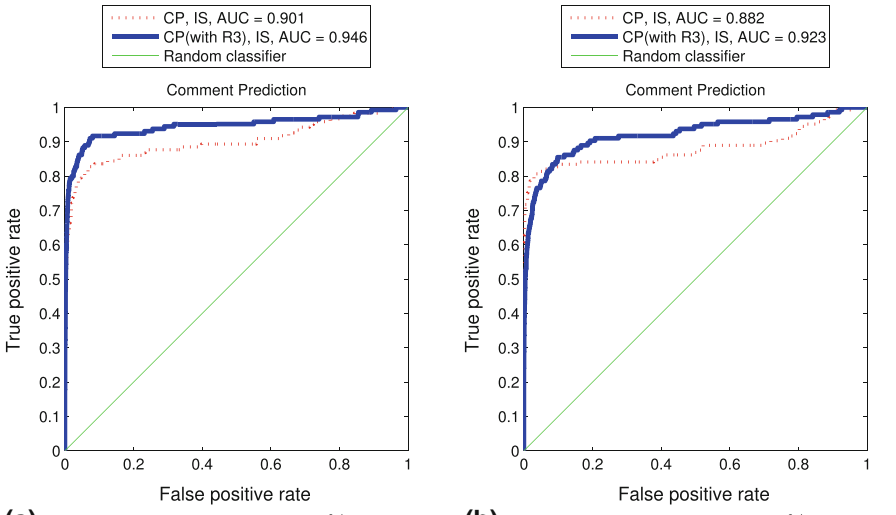


Fig. 13 Comparison of coupled models with different relations and IS cost on comment prediction of the models in Eqs. 31–33

we compare the average timings of the two models. CP requires ∼568 s and Tucker requires ∼2,654 s for comment prediction while CP requires ∼260 s and Tucker requires ∼1,435 s for Digg prediction in average.

In order to demonstrate the effect of various relational context on comment prediction, we also carried out experiments with different relational contexts. We observe that different combinations of the relations affect the prediction performance. Model of one of these combinations is given in Sect. 4.2. In this model, we incorporate the relation R3 with R1 and R2 to increase prediction performance on users' comment activities. Figure 13 shows the comparison of models given in Eqs. 26, 27 and 31–34 when 40 and 90 % of the data is unobserved, respectively.

Missing slice we test this case by setting randomly missing slices in user–story–comment tensor $\mathcal{X}_1$ for comment prediction task. Figure 14 demonstrates the performance of coupled models with IS divergence when 10- and 50 users' data are missing. Also note that CP is superior to Tucker as the amount of missing data increases.

Finally, we compare the prediction performances of our models in Eqs. 26, 27 and 37–40 under 40 and 90 % missing data in terms of P@10 metric. The results of Digg and comment predictions are given in Table 7.

Digg and comment prediction have also been studied in Lin et al. (2009) using the MFT (Metafac factorization with time evolving data) approach. The overall comment and Digg prediction performances of MFT algorithm were obtained as 0.135 ± 0.001 and 0.543 ± 0.007 in terms of P@10, respectively in Lin et al. (2009). While our prediction results are clearly higher than those of MFT in terms of P@10, we cannot directly compare them since we use the integrated Digg dataset instead of the segmented data and do not deal with the temporal aspect of the data. However, in Table 7, we observe that if we use the loss function and the tensor model used in MFT, i.e., CP model based on KL-divergence, then it performs worse than modeling the data

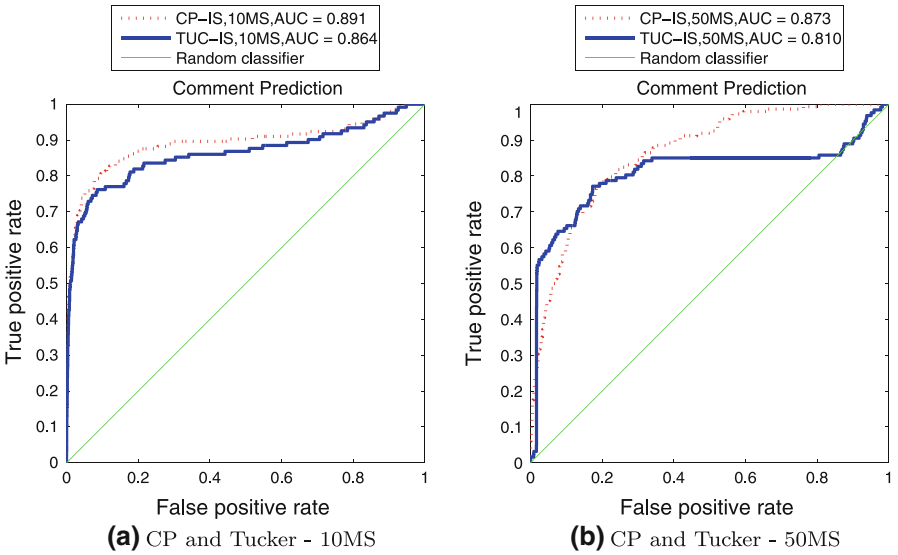


Fig. 14 Comment prediction result with missing slices and IS cost of the models in Eqs. 26, 27 for CP and Eqs. 29, 30 for Tucker

Table 7 The average prediction performance for Digg and comment prediction, evaluated by $P@10$ values, of the models in Eqs. 26, 27 and 37–40

Bold values indicate the best results

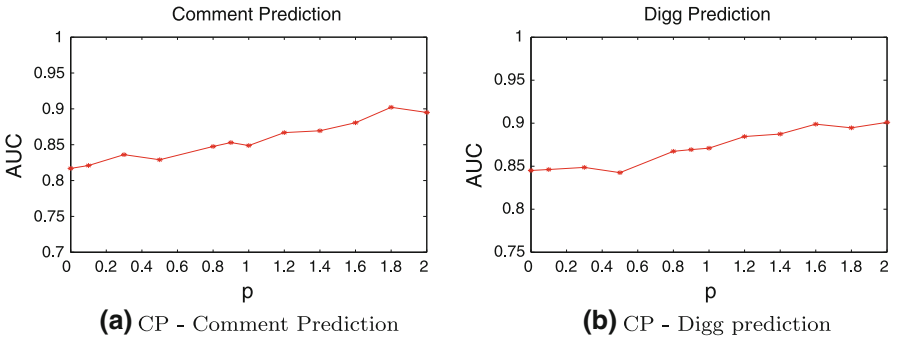
	Digg prediction		Comment prediction	
	40 %	90 %	40 %	90 %
GCTF–EUC	0.7154	0.6615	0.3632	0.3241
GCTF–KL	0.7414	0.6865	0.3751	0.3383
GCTF–IS	**0.7858**	**0.7479**	**0.4075**	**0.3846**

Table 8 Link prediction results on Digg with different experimental settings

	Digg prediction				Comment prediction			
	40 %		90 %		40 %		90 %	
	CP	Tucker	CP	Tucker	CP	Tucker	CP	Tucker
EUC	0.855	0.831	0.817	0.801	0.845	0.831	0.810	0.780
KL	0.921	0.882	0.853	0.827	0.871	0.845	0.824	0.810
IS	0.939	0.923	0.895	0.869	0.901	0.885	0.882	0.859

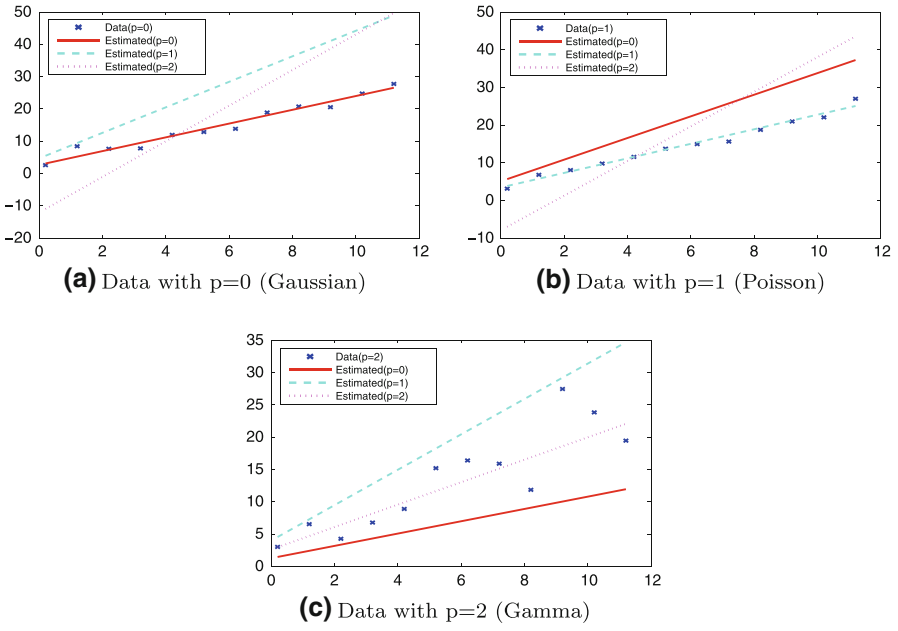
using IS-divergence. This clearly shows that GCTF framework is useful in terms of making use of better loss functions for modeling datasets. Table 8 summarizes the experimental results given in this section on Digg dataset in terms of AUC metric.

Effect of the cost function in addition, in order to demonstrate the effect of the cost function modeling the data, we have also carried out experiments on both coupled CP and Tucker models at different missing data fractions using different p values. Figure 15 illustrates the performance of arbitrary cost functions (with different p values) for the coupled CP model for both comment and Digg prediction when 90 % of the data is



(a) CP - Comment Prediction **(b)** CP - Digg prediction

Fig. 15 Comparison of different cost functions



(a) Data with p=0 (Gaussian) **(b)** Data with p=1 (Poisson)

(c) Data with p=2 (Gamma)

Fig. 16 Comparison of different p values for estimation of real data

unobserved. These results also confirm that IS-divergence, i.e., $p=2$, performs better than KL-divergence, i.e., $p=1$, which performs better than EUC distance, i.e., $p=0$.

Furthermore, in order to provide more insight regarding to the performance of different cost functions, we designed experiments on a synthetic dataset. Here, we generate data with different choices for p, that correspond to special cases of the exponential family distributions for any p-named Tweedie's family (Kaas 2005) such as the Gaussian ($p=0$), Poisson ($p=1$) and Gamma ($p=2$) distributions (Yılmaz and Cemgil 2012). These p values, i.e., $p = 0,\ 1,\ 2$, correspond to EUC, KL and IS cost functions, respectively. Then, we estimate the data by using all p values used for data generation. Figure 16 visualizes an example estimation. It can be observed that the best fit between the generated and estimated data occurs when we use the same p

value. These experiments demonstrate that the performance of the cost function is highly related to the distribution of data. For instance, IS-divergence performs better than EUC distance and KL-divergence for Digg dataset, so we can conclude that Digg is more likely to be distributed by a Gamma distribution.

6 Conclusions

In this article, we have studied link prediction problem using coupled analysis of relational data represented as datasets in the form of matrices and higher-order tensors. The problem is formulated as simultaneous factorization of higher-order tensors/matrices extracting common latent factors from the shared modes. While most existing studies on coupled analysis have been developed to fit a specific type of a tensor model using a particular loss function, we have used GCTF framework, which enables us to develop coupled models for joint analysis of multiple datasets in a compact way using various tensor models and cost functions. In our coupled analysis for the link prediction problem, in addition to the commonly-used KL-divergence and EUC distance-based loss functions, we have also studied IS-divergence as well as various other cost functions based on β-divergences.

Through extensive experiments on real datasets, we assess the performance of various alternative tensor models and loss functions for the link prediction problem. Numerical experiments clearly demonstrate that not only joint analysis of data from multiple sources via coupled factorization improves the link prediction performance but also the selection of right loss function and tensor model is crucial for accurately predicting missing links.

As a future direction and next step of this work, we aim to determine the relative weights of data included in the model, framed as dispersion parameter estimation; since the dispersion parameters play a key role on inference as they form a balance between the information obtained from multimodal observations.

Acknowledgments This work is funded by the TUBITAK Grant Number 110E292, Bayesian matrix and tensor factorisations (BAYTEN) and Boğaziçi University Research Fund BAP5723. It is also funded in part by the Danish Council for Independent Research—Technology and Production Sciences and Sapere Aude Program under the Projects 11-116328 and 11-120947.

Appendix

Computation for common factors

Here, we show the computation for A:

$$\Delta_{A,1}(\mathcal{Q}) = \left[\sum_{j,k} \mathcal{Q}^{i,j,k} \left(B^{j,r} C^{k,r} \right) \right] = Q_1(BC),$$

$$\Delta_{A,2}(\mathcal{Q}) = \left[\sum_{m} \mathcal{Q}^{i,m} \left(D^{m,r} \right) \right] = Q_2 D,$$

$$A \leftarrow A \circ \frac{Q_1(BC) + Q_2 D}{\hat{X}_1^{-p}(BC) + \hat{X}_2^{-p} D},$$

and B:

$$\Delta_{B,1}(Q) = \left[\sum_{i,k} Q^{i,j,k} \left(A^{i,r} C^{k,r} \right) \right] = Q_1(AC),$$

$$\Delta_{B,2}(Q) = \left[\sum_{n} Q^{j,n} \left(E^{n,r} \right) \right] = Q_2 E,$$

$$B \leftarrow B \circ \frac{Q_1(AC) + Q_3 E}{\hat{X}_1^{-p}(AC) + \hat{X}_3^{-p} E},$$

given in *Model 1*, Sect. 4.1.

Computational complexity

We have conducted experiments on tensor completion problem to demonstrate that time complexity of the modeling framework is $O(N)$ for sparse datasets, where N is the number of known entries. We consider two situations in these experiments: (i) $500 \times 500 \times 500$ three-way array with 99 % missing data (1.25 million known values), and (ii) $1,000 \times 1,000 \times 1,000$ three-way array with 98 % missing data (20 million known values). We have used CP tensor factorization model with $R = 3$ components to generate data, then added 20 % random Gaussian noise. We have then fitted a CP model using EUC distance-based loss function and used the extracted CP factors to reconstruct the data. Figure 17 shows the *average tensor completion performance of 10 independent runs* in terms of RMSE score. In the $500 \times 500 \times 500$ case, all ten problems have been solved with an RMSE score around 0.20, with computation times ranging between 400 and 500 s and in the $1,000 \times 1,000 \times 1,000$ case, all ten problems

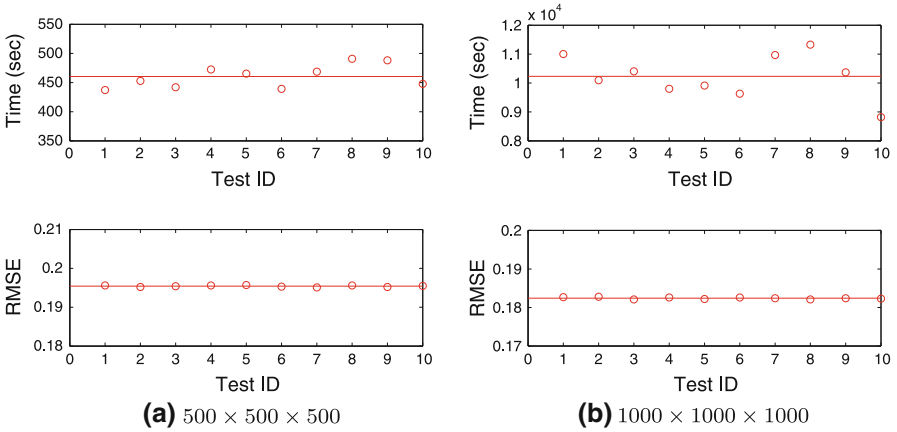


Fig. 17 Results of our algorithm for large-scale problems. The means are shown as *solid lines*

are also solved with an RMSE score around 0.20. The computation times have ranged from 8,000 to 12,000 s, approximately 20 times slower than the $500 \times 500 \times 500$ case, which has 16 times more non-missing entries.

References

Acar E, Kolda TG, Dunlavy DM (2011a) All-at-once optimization for coupled matrix and tensor factorizations. In: KDD'11 workshop proceedings

Acar E, Dunlavy D, Kolda TG, Morten M (2011b) Scalable tensor factorizations for incomplete data. Chemometr Intell Lab 106:41–56

Al Hasan M, Zaki MJ (2011) A survey of link prediction in social networks. In: Aggarwal CC (ed) Social network data analytics. Springer, New York

Alter O, Brown PO, Botstein D (2003) Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. Proc Natl Acad Sci USA 100:3351–3356

Banerjee A, Basu S, Merugu S (2007) Multi-way clustering on relation graphs. In: SDM'07, pp 145–156

Candès EJ, Plan Y (2010) Matrix completion with noise. Proc IEEE 98:925–936

Cao B, Liu NN, Yang Q (2010) Transfer learning for collective link prediction in multiple heterogenous domains. In: ICML'10, pp 159–166

Carroll JD, Chang JJ (1970) Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. Psychometrika 35:283–319

Choudhury MD, Sundaram H, John A, Seligmann DD (2009) Social synchrony: predicting mimicry of user actions in online social media. In: CSE, vol 4, pp 151–158

Cichocki A, Zdunek R, Phan AH, Amari S (2009) Nonnegative matrix and tensor factorization. Wiley, Chichester

Clauset A, Moore C, Newman M (2008) Hierarchical structure and the prediction of missing links in networks. Nature 453:98–101

Davis DA, Lichtenwalter R, Chawla NV (2011) Multi-relational link prediction in heterogeneous information networks. In: ASONAM'11, pp 281–288

Dunlavy DM, Kolda TG, Acar E (2011) Temporal link prediction using matrix and tensor factorizations. In: ACM TKDD'11, vol 5, Issue 2, Article 10

Ermis B, Cemgil AT (2013) A Bayesian tensor factorization model via variational inference for link prediction. In: NIPS 2013 workshop on probabilistic models for big data (PMBD)

Ermis B, Acar E, Cemgil TA (2012) Link prediction via generalized coupled tensor factorisation. In: ECML/PKDD workshop on collective learning and inference on structured data

Gandy S, Recht B, Yamada I (2011) Tensor completion and low-n-rank tensor recovery via convex optimization. Inverse Probl 27:025010

Getoor L, Diehl CP (2005) Link mining: a survey. ACM SIGKDD Explor Newsl 7(2):3–12

Harshman RA (1970) Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multi-modal factor analysis. UCLA Work Pap Phonetics 16:1–84

Harshman RA, Lundy ME (1996) Uniqueness proof for a family of models sharing features of Tucker's three-mode factor analysis and PARAFAC/candecomp. Psychometrika 61(1):133–154

Hitchcock FL (1927) Multiple invariants and generalized rank of a p-way matrix or tensor. J Math Phys 7:39–79

Jamali M, Lakshmanan L (2013) HeteroMF: recommendation in heterogeneous information networks using context dependent factor models. In: Proceedings of the 22nd international conference on World Wide Web, WWW '13, pp 643–654

Jiang M, Cui P, Liu R, Yang Q, Wang F, Zhu W, Yang S (2012) Social contextual recommendation. In: CIKM'12, pp 45–54

Kaas R (2005) Compound Poisson distributions and GLM's, Tweedie's distribution. Technical report. Royal Flemish Academy of Belgium for Science and the Arts, Brussels

Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37

Lin Y-R, Sun J, Castro P, Konuru R, Sundaram H, Kelliher A (2009) MetaFac: community discovery via relational hypergraph factorization. In: KDD'09, pp 527–536

Long B, Zhang (Mark) Z, Wu X, Yu PS (2006) Spectral clustering for multi-type relational data. In: ICML'06, pp 585–592

Ma H, Yang H, Lyu MR, King I (2008) Sorec: social recommendation using probabilistic matrix factorization. In: CIKM'08
Menon AK, Elkan C (2011) Link prediction via matrix factorization. In: ECML/PKDD'11, pp 437–452
Menon AK, Chitrapura KP, Garg S, Agarwal D, Kota N (2011) Response prediction using collaborative filtering with hierarchies and side-information. In: KDD'11, pp 141–149
Narita A, Hayashi K, Tomioka R, Kashima H (2011) Tensor factorization using auxiliary information. In: ECML PKDD'11, pp 501–516
Popescul A, Ungar LH (2003) Statistical relational learning for link prediction. In: IJCAI'03
Sanderson M (2010) Test collection based evaluation of information retrieval systems. Found Trends Inf Retr 4(4):247–375
Shi C, Kong X, Yu PS, Xie S, Wu B (2012) Relevance search in heterogeneous networks. In: EDBT. ACM, New York, NY, pp 180–191
Simsekli U, Cemgil AT (2012) Markov chain Monte Carlo inference for probabilistic latent tensor factorization. In: IEEE international workshop on machine learning for signal processing (MLSP)
Simsekli U, Cemgil AT, Yilmaz YK (2013a) Learning the beta-divergence in Tweedie compound Poisson matrix factorization models. In: Proceedings of the 30th international conference on machine learning (ICML-13), JMLR workshop and conference proceedings, May 2013, vol 28, pp 1409–1417
Şimşekli U, Ermiş B, Cemgil AT, Acar E (2013) Optimal weight learning for coupled tensor factorization with mixed divergences. In: EUSIPCO
Singh AP, Gordon GJ (2008) Relational learning via collective matrix factorization. In: KDD'08
Smilde AK, Westerhuis JA, Boque R (2000) Multiway multiblock component and covariates regression models. J Chemom 14:301–331
Spiegel S, Clausen JH, Albayrak S, Kunegis J (2011) Link prediction on evolving data using tensor factorization. In: PAKDD workshops, pp 100–110
Stäger M, Lukowicz P, Tröster G (2006) Dealing with class skew in context recognition. In: ICDCS workshops, p 58
Sun Y, Barber R, Gupta M, Aggarwal CC, Han J (2011) Co-author relationship prediction in heterogeneous bibliographic networks. In: ASONAM, pp 121–128
Tan VYF, Fevotte C (2013) Automatic relevance determination in nonnegative matrix factorization with the beta-divergence. IEEE Trans Pattern Anal Mach Intell 35(7):1592–1605
Taskar B, Wong M-F, Abbeel P, Koller D (2003) Link prediction in relational data. In: NIPS'03
Tucker LR (1963) Implications of factor analysis of three-way matrices for measurement of change. In: Harris CW (ed) Problems in measuring change. University of Wisconsin Press, Madison, pp 122–137
Tucker LR (1966) Some mathematical notes on three-mode factor analysis. Psychometrika 31:279–311
Wang C, Raina R, Fong D, Zhou D, Han J, Badros GJ (2011) Learning relevance from heterogeneous social network and its application in online targeting. In: SIGIR. ACM, New York, NY, pp 655–664
Yang S-H, Long B, Smola AJ, Sadagopan N, Zheng Z, Zha H (2011) Like like alike: joint friendship and interest propagation in social networks. In: WWW'11, pp 537–546
Yang Y, Chawla NV, Sun Y, Han J (2012) Predicting links in multi-relational and heterogeneous networks. In: ICDM'12, pp 755–764
Yilmaz YK (2012) Generalized tensor factorization. PhD Thesis, Bogazici University
Yilmaz YK, Cemgil AT (2010) Probabilistic latent tensor factorization. In: LVA/ICA, pp 346–353
Yılmaz YK, Cemgil AT (2012) Alpha/beta divergences and Tweedie models. arXiv: 1209.4280 v1
Yilmaz YK, Cemgil AT, Simsekli U (2011) Generalised coupled tensor factorisation. In: NIPS'11
Yoo J, Choi S (2012) Hierarchical variational Bayesian matrix co-factorization. In: ICASSP'12, pp 1901–1904
Yoo J, Kim M, Kang K, Choi S (2010) Nonnegative matrix partial co-factorization for drum source separation. In: ICASSP'10, pp 1942–1945
Yu X, Gu Q, Zhou M, Han J (2012) Citation prediction in heterogeneous bibliographic networks. In: SDM. SIAM/Omnipress, Anaheim, CA, pp 1119–1130
Zheng VW, Cao B, Zheng Y, Xie X, Yang Q (2010) Collaborative filtering meets mobile recommendation: a user-centered approach. In: AAAI'10
Zheng VW, Zheng Y, Xie X, Yang Q (2012) Towards mobile intelligence: learning from GPS history data for collaborative recommendation. Artif Intell 184–185:17–37