WTEN: An Advanced Coupled Tensor Factorization Strategy for Learning from Imbalanced Data

Quan Do[1], Thanh Pham[2], Wei Liu[1], and Kotagiri Ramamohanarao[2]

[1] Advanced Analytics Institute, University of Technology Sydney,
Sydney, Australia
{Quan.Do,Wei.Liu}@uts.edu.au,
[2] Department of Computing and Information Systems, University of Melbourne,
Melbourne, Australia
Thanhp1@student.unimelb.edu.au,Kotagiri@unimelb.edu.au

Abstract. *Learning from imbalanced and sparse data in multi-mode and high-dimensional tensor formats efficiently is a significant problem in data mining research. On one hand, Coupled Tensor Factorization (CTF) has become one of the most popular methods for joint analysis of heterogeneous sparse data generated from different sources. On the other hand, techniques such as sampling, cost-sensitive learning, etc. have been applied to many supervised learning models to handle imbalanced data. This research focuses on studying the effectiveness of combining advantages of both CTF and imbalanced data learning techniques for missing entry prediction, especially for entries with rare class labels. Importantly, we have also investigated the implication of joint analysis of the main tensor and extra information. One of our major goals is to design a robust weighting strategy for CTF to be able to not only effectively recover missing entries but also perform well when the entries are associated with imbalanced labels. Experiments on both real and synthetic datasets show that our approach outperforms existing CTF algorithms on imbalanced data.*

Keywords: Tensor Factorization; Coupled Tensor Factorization; Imbalanced data learning

1 Introduction

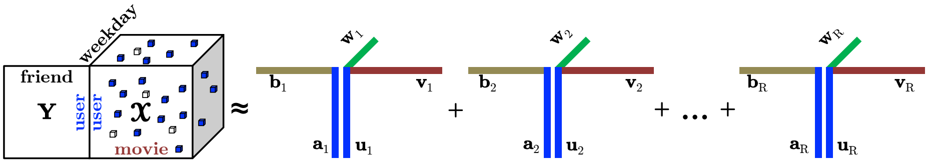
Recent innovations on the Internet and social media have made many multi-mode, high dimensional, sparse and imbalanced data available. Together with this explosive dimension growth, Coupled Tensor Factorization (CTF) has become one of the most popular methods for joint analysis of sparse data generated from different sources. It has also been proven to predict missing data entries with high accuracy [2]. In case the actual entries are skewed toward a particular class, generally, we want to achieve a high prediction rate of the class of interest in spite of its rarity. Nevertheless, even if the reconstructed tensor predicts everything to be of the majority class, the overall accuracy is still very high. For example, in the event of a binary class (such as high and low ratings of movies) and the actual entries are skewed towards a particular class, for instance, the ratio of negative class (e.g., low ratings of movies) to positive class (e.g., high ratings) is 99 to 1, any CTF would easily achieve 99 percent overall approximation accuracy by just approximating all missing entries to the negative class. This 99 percent precision rate is impressive enough if we ignore the 0 percent accuracy of the positive class. This bias accuracy not only reduces the robustness of the model but also might cause severe consequences, especially in cases that most of the observed samples are normal and just a few rare cases are anomaly ones. For example in disease diagnosing application, even though most of the training data are healthy specimens, predicting an unhealthy sample as a healthy one costs extremely high, in many cases a human life. However, it might be acceptable to classify a healthy person as an unhealthy and perform a few other diagnoses. Achieving a high rate on classifying the rare case without jeopardizing the majority class is, therefore, a crucial requirement in this instance.

Learning from imbalanced data has attracted considerable attention in knowledge discovery community. Sampling [7] and cost-sensitive [8] approaches have been studied to deal with imbalanced datasets. Although they have been proposed to decision trees and neural networks, they have not yet been applied to multi-mode, high dimensional, sparse and imbalanced tensor data (and importantly, its decompositions). This significant theoretical gap motivates us to take the advantages of both CTF and imbalanced data learning techniques to address the problem of recommending missing entries from imbalanced yet sparse heterogeneous datasets. In particular, we adjust CTF's objective function by a weighting strategy that lowers the significance of wrongly recommending the majority class and strengthens the importance of correctly estimating the rare case. Here we introduce a weighting strategy for CTF, called WTEN, as the first CTF approach for imbalanced data learning. Although this paper targets binary missing label estimation problem, the weighting strategy can be straightforwardly applied to multiple labels, such as integer ratings where the frequencies of integers are imbalanced.

In brief, our main contribution in this paper are the following:

1) **Performance**: we propose a novel weighting strategy, named WTEN (Weighted Tensor Factorization), for missing entry recommendation using CTF. Our model robustly assigns effective weights with respect to different classes' approximation, and consequently performs significantly better on the minority class estimation without jeopardizing the majority one. WTEN is the first method, to our best knowledge, that enables CTF to handle imbalanced missing data entries.

2) **Foundation**: we study the effectiveness of joint analysis of the main tensor and the additional coupled data in CTF techniques for handling sparsity and imbalance over Tensor Factorization. Our theoretical analysis and experimental results suggest CTF to serve as a foundation for a general purpose latent factor imbalanced data learning.



(a) (b)

Fig. 1. Factorization of coupled imcomplete data sets for missing entries recovery. a) Correlation among different aspects of a dataset. $\boldsymbol{\mathcal{X}}$ is a tensor of ratings made by users for movies on weekdays. Dark boxes are observed low ratings (which are majority) and white boxes are known minority high ratings. Matrix $\mathbf{Y}$ represents user information. Movie rating tensor $\boldsymbol{\mathcal{X}}$ is, therefore, coupled with user information matrix $\mathbf{Y}$ in 'user' mode. b) $\boldsymbol{\mathcal{X}}$ is factorized as a sum of low rank factors that can be used to recover missing majority as well as minority cases.

3) **Usability and reproducibility**: Our factorization method with weighting scheme can be easily extended to different datasets and applications. Performance of WTEN is validated by both real-world and synthetic datasets. To promote the reusability of our idea, we open our source code with this paper.[3]

The rest of this paper is organized as follows. We introduce the background of tensor factorization in Section 2 followed by a review of existing work in Section 3. Section 4 explains our proposed idea. Experimental results together with our discussion are included in Section 5. We finally conclude our work in Section 6.

2 Preliminary

This section provides a brief introduction of core definitions and preliminary concepts of tensor, tensor factorization and coupled tensor factorization.

2.1 Tensor and our notations

Tensors are multidimensional arrays which are often specified by their number of modes (a.k.a., orders or ways). In specific, a mode-1 tensor is a vector; a matrix is a mode-2 tensor. A mode-3 or higher-order tensor is often called tensor in short. We denote tensors by boldface Euler script letters, e.g. $\boldsymbol{\mathcal{X}}$. We use boldface capitals, e.g. $\mathbf{A}$, for matrices. A boldface Euler script with indices in its subscript is used for an entry of a tensor while a boldface capital with indices in its subscript is for an entry of a matrix. For example, $\mathbf{A}_{\mathrm{i,j}}$ is an entry at row i and column j of matrix $\mathbf{A}$; the $(\mathrm{i,j,k})^{\mathrm{th}}$ entry of $\boldsymbol{\mathcal{X}}$ is $\boldsymbol{\mathcal{X}}_{\mathrm{i,j,k}}$. Table 1 lists all the symbols we throughout use in this paper.

[3] Our source code is available at `https://github.com/quanie/WTEN`

Table 1. Symbols and their description

Symbol	**Description**
$\mathcal{X}$	A tensor
$\mathbf{X}$	A matrix
$\mathcal{X}_{i,j,k}$	An entry of a tensor
$\mathbf{X}_{i,j}$	An entry of a matrix
$\hat{\mathcal{X}}_{i,j,k}$	A reconstructed missing entry of tensor $\mathcal{X}$
$\|\mathbf{A}\|_F$	Frobenius norm
$\mathbf{U}^{(n)}$	A n-th mode factor
I*J*K	Dimensions of tensor $\mathcal{X}$
I*M	Dimensions of matrix Y
R	Decomposition rank
$\circ$	Khatri-Rao product
$\circledast$	Hadamard (elementwise) product
$\mathcal{L}$	Loss function

2.2 Tensor Factorization (TF) and Coupled Tensor Factorization (CTF)

Tensor factorization, based on PARAFAC decomposition [11], approximates a high-order tensor into a sum of a finite number of low rank factors.

$$\mathcal{X} \approx \sum_{r=1}^{R} \prod_{n=1}^{N} \mathbf{U}^{(n)}_{I_n,r}$$

where $\mathcal{X} \in \mathbb{R}^{I_1 * I_2 * \cdots * I_N}$ is a N-mode tensor and its N rank-R factors are $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n * R}, \forall n \in [1, N]$.

The goal of PARAFAC decomposition is to find the best low-dimensional approximation of $\mathcal{X}$ [14]. In other words, PARAFAC decomposition finds

$$\min_{\hat{\mathcal{X}}} \|\mathcal{X} - \hat{\mathcal{X}}\|_F \quad \text{with} \quad \hat{\mathcal{X}} = \sum_{r=1}^{R} \prod_{n=1}^{N} \mathbf{U}^{(n)}_{I_n,r}$$

and $\mathcal{L} = \|\mathcal{X} - \hat{\mathcal{X}}\|_F$ is defined as the loss function of the factorization.

In case $\mathcal{X}$ is a mode-3 tensor, TF decomposes $\mathcal{X}$ into a Khatri-Rao product of its factors, and thus the loss function is defined by:

$$\mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \frac{1}{2} * \|\mathcal{X} - \mathbf{U} \circ \mathbf{V} \circ \mathbf{W}\|_F^2$$

where $\mathbf{U} \in \mathbb{R}^{I*R}$, $\mathbf{V} \in \mathbb{R}^{J*R}$ and $\mathbf{W} \in \mathbb{R}^{K*R}$.

We often have additional information in a format of a matrix or a tensor which has one or more modes in common with the main tensor. These side information along with the main data can help to deepen our understanding of the underlying patterns in the data, and to improve the accuracy of tensors composition. For example, Acar et al. [2] defined an objective function for joint analysis of a tensor $\mathcal{X}$ coupled with a matrix $\mathbf{Y}$ in its first mode by:

$$\mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{W}, \mathbf{A}) = \frac{1}{2} * \|\mathcal{X} - \mathbf{U} \circ \mathbf{V} \circ \mathbf{W}\|_{\mathrm{F}}^2 + \frac{1}{2} * \|\mathbf{Y} - \mathbf{U}\mathbf{A}^{\mathrm{T}}\|_{\mathrm{F}}^2 \tag{1}$$

where $\mathbf{U}$ is the common factor of both $\mathcal{X}$ and $\mathbf{Y}$.

By solving this equation (1) with an optimizer, low rank factors $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{A}$ can be obtained. These factors can then be used to approximate both tensor $\mathcal{X}$ and matrix $\mathbf{Y}$.

2.3 Missing data completion

Latent factors discovered by TF or CTF above can be used to recover missing data from the original input tensor. The most widely used approach [2] is to utilize these latent factors to reconstruct $\hat{\mathcal{X}}$ for missing entries recovery. Suppose a tensor $\mathcal{X}$ coupled with a matrix $\mathbf{Y}$ are factorized by (1). A missing entry (i,j,k) of $\mathcal{X}$ is estimated by:

$$\hat{\mathcal{X}}_{\mathrm{i,j,k}} = \sum_{\mathrm{r=1}}^{\mathrm{R}} \mathbf{U}_{\mathrm{i,r}} \mathbf{V}_{\mathrm{j,r}} \mathbf{W}_{\mathrm{k,r}} \tag{2}$$

In the event of binary entries, a simple method to decide a label of $\hat{\mathcal{X}}_{\mathrm{i,j,k}}$ is to use a threshold ϵ. An entry (i,j,k) of $\mathcal{X}$ belongs to negative label if $\hat{\mathcal{X}}_{\mathrm{i,j,k}} \leq \epsilon$ or else positive label.

3 Literature Review

Learning from imbalanced data has attracted considerable attention in knowledge discovery community. Imbalanced data learning algorithms proposed in the literature can be categorized into sampling, cost sensitive, kernel based approaches [12]. Kernel based methods that mainly focus on modifying SVM kernel for imbalanced learning [25] or applying sampling to SVM framework [3] are out of scope of this paper. In this section, we provide a brief overview of sampling and cost sensitive methods.

When a training data is skewed toward a particular class, a straightforward strategy [7] is sampling to create a more balanced data distribution for both classes. Two sampling techniques, oversampling and undersampling, are widely proposed for imbalanced learning. Oversampling increases the minority class population by creating more data samples. The extra data samples can be made by replicating minority samples [13] or synthesized by various techniques such as Synthetic Minority Oversampling Technique (SMOTE) [6]. Overfiting is often considered a potential disadvantage of oversampling [7]. Undersampling, on the other hand, reduces the majority class by eliminating some of its samples. This reduction can be done randomly [20] or based on statistical knowledge [19]. Both randomly and statistically undersampling have the possibility of losing important data.

An alternative method to overcome data imbalance is a cost-based approach. Instead of balancing the data distribution by sampling, cost sensitive learning [8] associates different penalties with misclassifying different classes correspondingly. For example, in case a training data is skewed towards negative label, the total cost of misclassifying negative classes as positive ones outweighs that of misclassifying positive labels as negative. Any learning algorithm that minimizes total misclassification cost mostly optimizes the negative class only [23]. By associating a higher cost with misclassifying a positive class as a negative one than with the contrary, the algorithm now balances better for the positive class. As a result, this cost-sensitive approach improves the classification performance with respect to the rare class. Although this technique has been successfully applied to decision tree via subtree pruning [5] or data split [16], and neural networks [26], it has not ever been proposed for high-dimensional decompositions of tensors with imbalanced data entries.

Tensor factorization (TF) has been used for multi-mode, high-dimensional and sparse data analysis with a goal to capture the underlying low rank structures. This analysis has become a new trend since the Netflix Prize competition [15] where it is used to predict movie ratings with high accuracy. Researchers has extended TF to do joint data analysis. Early work by Singh and Gordon [24] introduced Collective Matrix Factorization (CMF) to take an advantage of correlations between different coupled matrices and simultaneously factorized them. CMF techniques have been successfully applied to capture the underlying complex structure of data [15, 21]. Acar, Kolda and Dunlavy [2] later expanded CMF to CTF to handle Coupled Matrix and Tensor Factorization by modeling heterogeneous data sets as higher-order tensors and matrices in a coupled loss function. They also proved the possibility of using these low rank factors to recover missing entries. Tensor methods have been studied for factorization with labeled information [17] and also compression with tensor representations [18]. Papalexakis et al. [22] and Beutel et al. [4] scaled CTF up to parallel and distributed environments but with the same loss function as proposed by other authors.

Motivations for this research

Despite the popularity of CTF on high-dimensional datasets, improvements of CTF on imbalanced data has not been studied. If we apply CTF with its traditional objective function (1) on imbalanced data, it will tend to ignore the minority cases and approximate most missing values to be the majority ones. The objective function is still optimal thanks to the fact that almost all predicted instances are correct. This is because the loss function (1) assumes that errors of factorizing the majority class in $\mathcal{X}$ and that of decomposing the minority one contribute equally to the final loss of the CTF. Apparently, this is not the case for imbalanced data as the majority class extremely out-represents the minority one. Thus, the loss of factorizing the majority class totally outweighs all the loss of decomposing the rare one. The algorithms, hence, focus on optimizing the major class to achieve a lower loss.

Another problem with CTF on imbalanced learning is that both oversampling and undersampling do not work effectively. First of all, oversampling does not balance out the data distribution. Suppose $\mathcal{X}_{\mathrm{i,j,k}}$ is an entry of the minority class, oversampling by duplicating $\mathcal{X}_{\mathrm{i,j,k}}$ does not add anything to the tensor as an entry at the index {i,j,k} is already there in $\mathcal{X}$. Hence, the data distribution does not change. Secondly, even though undersampling creates a more balanced data distribution, it may remove some important observed data. This is especially critical as the observed data is sparse. Losing more data might prevent CTF from achieving its optimization, thus, reducing its accuracy. Last but not least, sampling cannot straightforwardly be done on the additional data in a form of coupled matrices or tensors, and doing so again does not make any change on the data distribution of the main tensor.

Motivated by the above significant theoretical gaps, in this paper we propose a novel cost-sensitive weighting strategy to overcome the imbalanced data problem in high dimensional and heterogeneous datasets. Our algorithm optimizes the factorization of both the majority and minority class in a balanced manner, significantly improving missing entry estimations of the minority class.

4 WTEN: Weighted Tensor Factorization for Imbalanced Data

In this section, we introduce our proposed WTEN to handle imbalanced datasets. Suppose $\mathcal{X} \in \mathbb{R}^{\mathrm{I*J*K}}$ is a mode-3 tensor coupled with a matrix $\mathbf{Y} \in \mathbb{R}^{\mathrm{I*M}}$ in their first mode, and suppose their data entries are binary. In an event when $\mathcal{X}$ is skewed toward class 0, algorithms [2, 4, 22] with the objective function (1) show their drawbacks in estimating class 1 as they approximate everything to be of the majority class. Yet the objective function is still considered as optimal because almost all predicted instances are correct. This hence reduces the robustness of the methods in dealing with the imbalanced input.

One of a few possible improvements of the above problem is to properly highlight the impact of errors in approximating the rare cases. This can be done by adjusting the objective function (1) to a weighted version based on observed frequencies of different classes. So the objective function (1) becomes:

$$\begin{aligned} \mathcal{L}(\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{A}) = & \mathrm{w}_0 * \|\mathcal{X}_0 - \mathbf{U} \circ \mathbf{V} \circ \mathbf{W}\|_{\mathrm{F}}^2 + \mathrm{w}_1 * \|\mathcal{X}_1 - \mathbf{U} \circ \mathbf{V} \circ \mathbf{W}\|_{\mathrm{F}}^2 \\ & + \|\mathbf{Y} - \mathbf{U}\mathbf{A}^{\mathrm{T}}\|_{\mathrm{F}}^2 \end{aligned} \tag{3}$$

where $\mathcal{X}_0$ and $\mathcal{X}_1$ are tensor entries containing negative and positive labels, respectively; w_0 is a weight of precisely estimating class 0 and w_1 is that of correctly approximating class 1. The first term represents prediction error of class 0 whereas the second term captures that of class 1.

If appropriate weights are used, w_0 and w_1 will have an effect of balancing out the impact of misclassifying different classes with respect to their observation ratio. An effective approach is to assign $\mathrm{w}_0 = \mathrm{N}_1/\mathrm{size}(\mathcal{X})$ and $\mathrm{w}_1 = \mathrm{N}_0/\mathrm{size}(\mathcal{X})$ where N_0 and N_1 are the number of observed 0s and 1s in $\mathcal{X}$, respectively, and $\mathrm{size}(\mathcal{X})$ denotes the number of observed elements of $\mathcal{X}$. Doing so lowers importance of approximating the majority class. Thus, this weighting strategy will likely improve the estimation of the minority label.

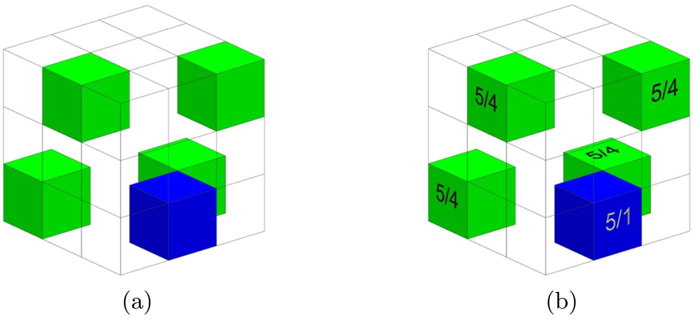
The objective function (1) can also be adjusted by a weighting tensor as the following:

$$\mathcal{L}(\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{A},\mathbf{B}) = \|\mathcal{W} \circledast (\mathcal{X} - \mathbf{U} \circ \mathbf{V} \circ \mathbf{W})\|_{\mathrm{F}}^2 + \|\mathbf{Y} - \mathbf{U}\mathbf{A}^{\mathrm{T}}\|_{\mathrm{F}}^2 \tag{4}$$

where $\mathcal{A} \circledast \mathcal{B}$ is a Hadamard (element-wise) product of $\mathcal{A}$ and $\mathcal{B}$ which yeilds a tensor $\mathcal{C}$ with entries $\mathcal{C}_{\mathrm{i,j,k}} = \mathcal{A}_{\mathrm{i,j,k}} * \mathcal{B}_{\mathrm{i,j,k}}$ and $\mathcal{W}$ is a weighting tensor having the same size of $\mathcal{X}$, but its entries' value is determined by

$$\mathcal{W}_{\mathrm{i,j,k}} = \begin{cases} \frac{\mathrm{size}(\mathcal{X})}{\mathrm{N}_0} & \text{when } \mathcal{X}_{\mathrm{i,j,k}} = 0 \\ \frac{\mathrm{size}(\mathcal{X})}{\mathrm{N}_1} & \text{when } \mathcal{X}_{\mathrm{i,j,k}} = 1 \end{cases}$$

This weighting tensor, $\mathcal{W}$ as illustrated in Figure 2, will have an effect of increasing the impact of errors in approximating the minority class. This weighting scheme produces the same result as the approach suggested in (3). Yet, an implementation of (4) might be simpler as its gradient is likely to be more straightforward to be computed in the optimization processes.



(a) (b)

Fig. 2. Imbalanced and sparse tensor of size $\mathrm{R}^{3\times3\times3}$, and its weighting tensor. a) Data tensor $\mathcal{X}$ where minority class, majority class and missing entries are represented by blue (darker), green (lighter) and transparent boxes, respectively. b) Weighting tensor $\mathcal{W}$ where each observed entries of $\mathcal{X}$ will be assigned a weight (5/4 for the majority and 5/1 for the minority).

Equation (3) as well as (4) overcomes the problem of the conventional loss function (1) in dealing with imbalanced data. By introducing different weighting parameters, they balance out importances of predicting both the majority class and the minority one. In other words, this non-uniform weighting strategy either emphasizes the impact of errors in estimating the minority class or reduces the significance of losses in approximating the majority case so that WTEN is very well balanced in optimizing the performance of both class labels.

5 Performance Evaluation

Our goals of conducting experiments below are to assess: 1) the contribution of an additional coupled matrix to the tensor factorization accuracy and 2) the effectiveness of our proposed weighting strategy on the estimation of missing entries when it is applied to imbalanced datasets. For better validating our work, both synthetic datasets in which the imbalance rate and coupled relationship are controlled and two real-world data are used.

5.1 Data used in our experiments

We use two synthetic and two real-world datasets for our experiments. The following subsections explain how synthetic data is generated and introduce two real-world datasets.

5.1.1 Synthetic data

Two datasets are synthesized by the following steps:

- Step1: A symmetric user-by-user matrix $\mathbf{Y_{50}}$ with 0s and 1s is randomly created to represent friendship among users. Value 1 means a pair of users is friend, value 0 means otherwise. Each user has a certain percentage of all users as friends. In this experiment, we randomly generated $\mathbf{Y_{50}}$ with about 50% of all users as friends.

- Step 2: For every user, a set of random ratings of 1s (for 5-star ratings) and 0s (other ratings) for movies over twelve months is generated following a rule that ensures any pair of users with 1 in $\mathbf{Y_{50}}$ has almost the same rating patterns. This is to capture the fact that users who are friends usually have similar preferences for movies over the year. The generated ratings are in a tensor format of (users, movies, months). Two different sparse tensors are synthesized for this experiment with the ratio between 0 and 1 ratings of 100:1 ($\mathcal{X}_0$) and 1,000:1 ($\mathcal{X}_1$) as summarized in Table 2.

Table 2. Ground truth distributions of the two synthesized tensors $\mathcal{X}_0$ and $\mathcal{X}_1$ of size 100 x 100 x 12, a real-world ABS tensor $\mathcal{X}_2$ of size 153 x 88 x 3 and a real-world MovieLens tensor $\mathcal{X}_3$ of size 943 x 1,682 x 7

Label	**$\mathcal{X}_0$ (100:1)**		**$\mathcal{X}_1$ (1,000:1)**		**$\mathcal{X}_2$ (9:1)**		**$\mathcal{X}_3$ (4:1)**	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
1	192	48	19	5	664	167	16,744	4,457
0	19,200	4,800	19,200	4,800	5,799	1450	63,256	15,543

- Final step: just like real-world scenarios where users make friends with those who have similar preferences or unfriend those who do not while their ratings in the past do not change, we analyze each of the two tensors to find pairs of users having the same rating patterns. 1s are then added to these pairs in $\mathbf{Y_{50}}$ to create other two matrices $\mathbf{Y^0_{80}}$ and $\mathbf{Y^1_{80}}$ of about 80% of friendship for each tensors, respectively. The same process is done to form other two matrices $\mathbf{Y^0_{20}}$ and $\mathbf{Y^1_{20}}$ of about 20% of friendship by removing 1s in $\mathbf{Y_{50}}$ for pairs of users having unique rating patterns. These relationships are showed in Table 3.

Table 3. Matrices coupled with synthetic tensors $\mathcal{X}_0$ and $\mathcal{X}_1$ for CTF.

Tensor	**Matrix**		
	Friendship rate		
	20%	50%	80%
$\mathcal{X}_0$	$\mathbf{Y^0_{20}}$	$\mathbf{Y_{50}}$	$\mathbf{Y^0_{80}}$
$\mathcal{X}_1$	$\mathbf{Y^1_{20}}$	$\mathbf{Y_{50}}$	$\mathbf{Y^1_{80}}$

5.1.2 ABS data

Australian Bureau of Statistics (ABS) [1] publishes a comprehensive data about people and families for all Australia geographic areas. This ABS dataset has income ranges of different family types within 153 New South Wales' areas, so-called "local government areas", in 2001, 2006 and 2011, forming a tensor $\mathcal{X}_2$ of (area, income range, year) of size 153 by 88 by 3. $\mathcal{X}_2$ has 8080 observations whose values are 1s for nontrivial income ranges and 0s for trivial ones. ABS dataset also includes population, number of services provided, and Socio-Economic Indexes for Areas that rank areas with respect to their relative socio-economic advantage and disadvantage. This additional information is compiled into a 153 by 3 matrix $\mathbf{Y}_2$ of (area, profile). In this paper, we train our model with 80% of known $\mathcal{X}_2$'s entries, together with a fully observed $\mathbf{Y}_2$. The rest 20% of known entries of $\mathcal{X}_2$ are for testing. Table 2 summarizes this ABS data distribution.

5.1.3 MovieLens data

MovieLens dataset [10] includes ratings from 943 users for 1,682 movies. It is compiled into tensor $\mathcal{X}_3$ of (users, movies, weekdays) whose entries are ratings, matrix $\mathbf{Y}_3$ of (users, users' profile) and matrix $\mathbf{Z}_3$ of (movies, genres). Matrix $\mathbf{Y}_3$ has the size of 943 by 83 in which a user is specified by her gender (0 or 1), is grouped in one of 61 age groups, and have one of 21 occupations. Matrix $\mathbf{Z}_3$ categories 1,682 movies into 19 different genres. One movie belongs to one or more genres. Finally, values of $\mathcal{X}_3$'s entries are 1s for high ratings (e.g. 5-star) and 0s for observed low ratings (e.g. 1-star to 4-star). In this paper, we train our model with 80,000 known ratings, together with $\mathbf{Y}_3$ and $\mathbf{Z}_3$ of 2,159 and 2,893 observed nonzeros, respectively. 20,000 ratings are for testing. MovieLens data distribution is also shown in Table 2.

5.2 Factorization Accuracy

We investigate effects of an additional matrix $\mathbf{Y}$ to a tensor $\mathcal{X}$'s factorization accuracy by comparing the performance of decomposing only $\mathcal{X}$ (TF in this case) and that of joint factorizing $\mathcal{X}$ and $\mathbf{Y}$ (CTF whose coupled relationships are defined in Table 3). Both mean squared errors (MSE) (5) of the training sets and approximation results of the testing sets are the metrics for our evaluation.

$$\mathrm{MSE} = \frac{\|\mathcal{X} - \mathbf{U} \circ \mathbf{V} \circ \mathbf{W}\|_{\mathrm{F}}^2}{\mathrm{size}(\mathcal{X})} \tag{5}$$

where size($\mathcal{X}$) denotes the size of tensor $\mathcal{X}$. In case $\mathcal{X}$ is a sparse tensor, it is the number of observed elements of $\mathcal{X}$.

By having additional data in a form of coupled matrix, CTF improves the factorization accuracy of the main tensor over TF as illustrated in Figure 3 and Figure 4. As one may anticipate, having additional information in the event of extremely skewness towards one class is very crucial. Figure 3b shows the additional matrices help improve the MSE of factorizing $\mathcal{X}_1$ 60 times on average compared with factorizing $\mathcal{X}_1$ alone. The more interesting points lie in Figure 3a where the lower friendship rate, in other words, less informative, a coupled matrix is, the lower training MSE of factorizing $\mathcal{X}_0$ achieves. In particular, as information richness increases from left to right ($\mathbf{Y_{20}^0}$, $\mathbf{Y_{50}}$, then $\mathbf{Y_{80}^0}$), the MSEs of $\mathcal{X}_0$ when joint factorizing $\mathcal{X}_0$ with $\mathbf{Y_{20}^0}$, $\mathbf{Y_{50}}$ and $\mathbf{Y_{80}^0}$ raise correspondingly, even to higher than that of decomposing tensor $\mathcal{X}_0$ alone. This does not mean joint factorizing a tensor $\mathcal{X}_0$ with a stronger constraint and more informative matrix performs worse. Actually, a stronger constraint and more essential user-user information in $\mathbf{Y_{80}^0}$ guides the factorization of $\mathcal{X}_0$ towards a resistant of the conventional trend that approximates everything to be the majority class label to achieve a better estimation of the minority case. This resistance, thus, increases the training MSEs.

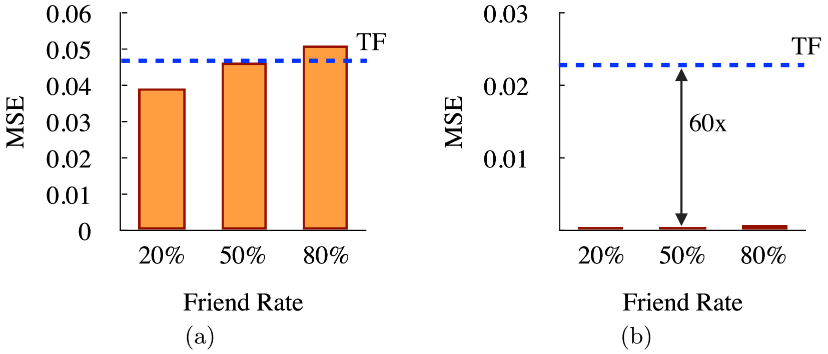


Fig. 3. MSE of a) $\mathcal{X}_0$ and b) $\mathcal{X}_1$ when they are joint factorized with different matrices. Reference lines (blue) in a) and b) are MSEs of decomposing $\mathcal{X}_0$ and $\mathcal{X}_1$ alone (by TF).

CTF with coupled richer information matrices enables more tested minority class (class 1) to be correctly approximated. As illustrated in Figure 4a, many more tested minority class is correctly recovered in case 80% friendship matrix $\mathbf{Y^0_{80}}$ is coupled factorized with $\mathfrak{X}_0$. When $\mathbf{Y_{50}}$ is used, CTF has similar performance with TF since information in $\mathbf{Y_{50}}$ already includes in $\mathfrak{X}_0$ which has been done in the second step of generating synthetic data. So $\mathbf{Y_{50}}$ does not really add any extra information to $\mathfrak{X}_0$ decomposition. Less informative $\mathbf{Y^0_{20}}$ has less meaningful information, compared to the other two matrices, to guide CTF toward correct direction, hence, performs worst. The same trend observed in Figure 4b with $\mathfrak{X}_1$ suggests a dominance of CTF over TF when an extra and meaningful matrix is joint decomposed with a tensor.

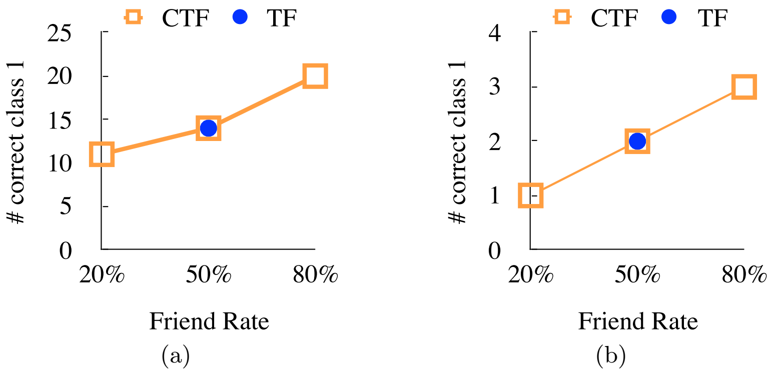


Fig. 4. Number of correctly approximation of missing 1s in a) $\mathfrak{X}_0$ and b) $\mathfrak{X}_1$. In both cases, the richer additional data is joint decomposition, the higher prediction rate is achieved.

5.3 Missing entry recovery

We compare our proposed WTEN with existing CMTF-OPT [2] and Sampling CMTF in which imbalanced data is first randomly under-sampling and then factorized by CMTF-OPT on missing entry recovery. Our target is to assess how well these algorithms approximate missing entries of the imbalanced ABS and MovieLens tensors. A missing entry (i,j,k) of $\mathfrak{X}$ is classified as 0 (a majority or a negative label) if the reconstructed $\hat{\mathfrak{X}}_{\mathrm{i,j,k}} \leq \epsilon$ or 1 (a minority or a positive label) otherwise. Recall, Precision and the area under a ROC curve [9] (AUC) which are widely used metrics in imbalanced data learning are our measurements. CMTF-OPT is optimized by three different optimization methods including Nonlinear Conjugate Gradient (NCG), Limited-memory BFGS (LBFGS) and Truncated Newton (TN) whereas WTEN is optimized by Stochastic Gradient Descent.

Table 4 summarizes the result of CMTF-OPT, Sampling CMTF and our proposed WTEN on missing imbalanced data recovery for both ABS dataset

Table 4. Performance of missing entries estimation with real-world ABS and MovieLens datasets. In both cases, WTEN achieves the highest accuracy on positive labels.

Algorithms	**ABS dataset**			**MovieLens dataset**		
	Precision	Recall	AUC	Precision	Recall	AUC
CMTF-OPT (NCG)	0.7658	0.7246	0.8495	**0.5477**	0.3143	0.6199
CMTF-OPT (LBFGS)	0.7602	0.7784	0.8751	0.4512	0.3330	0.6084
CMTF-OPT (TN)	**0.7697**	0.7605	0.8671	0.4372	0.2165	0.5683
Sampling CMTF (NCG)	0.4139	0.8922	0.8733	0.3653	0.7247	0.6818
Sampling CMTF (LBFGS)	0.3495	0.8623	0.8387	0.3249	0.4451	0.5900
Sampling CMTF (TN)	0.3830	0.8623	0.8511	0.3468	0.6679	0.6536
WTEN	0.4825	**0.9102**	**0.8989**	0.4055	**0.7393**	**0.7142**

and MovieLens dataset. Boldface numbers highlight the best among the algorithms for each dataset. As shown in Table 4, CMTF-OPT produces the highest Precision for both datasets as it approximates most of the tested entries to the majority labels (e.g. 0s), leading to a low false positive rate. This is confirmed for both datasets by CMTF-OPT's lowest Recall measurements, which denote the percentage of correctly estimating the minority labels (e.g. 1s), compared to the others. Sampling CMTFs with different optimization methods improve CMTF-OPT's performance on imbalanced data with higher Recalls. However, their performances are outweighed by WTEN which accurately estimates the positive labels even more (shown by the highest Recall in both cases) without jeopardizing the negative ones (illustrated by just a little lower Precision compared with the best CMTF-OPT). AUC also confirms the dominance of WTEN over existing algorithms on imbalanced data as WTEN achieves the highest AUC for both ABS and MovieLens datasets. All of these results demonstrate the performance of missing entry recovery on imbalanced data does not improve significantly by using a more sophisticated optimizer or applying sampling on the input imbalanced data, but in fact, our proposed strategy enables CTF to achieve a better performance.

To illustrate the advantage of our proposed WTEN, we present in Figure 5 the convergence of all the algorithms on MovieLens training data. There are two insights we can observe in this figure. Firstly, CMTF-OPTs' least squares errors are generally lower than that of WTEN. This is because WTEN decomposes the input tensor with respect to both majority and minority labels optimization, whereas CMTF-OPTs focuses on minimizing the lost of the majority ones leading to lower least squares errors. Secondly, the convergence speed of WTEN is the same as, if not better than, different optimizers of CMTF-OPT. They almost reach the optimum after about 10 seconds. This convergence evidence together with WTEN's enhanced performance of missing entries on both real-world datasets confirms its significance on improving the accuracy of minor class estimation, suggesting WTEN as the most appropriate method for CTF to handle imbalanced data learning.

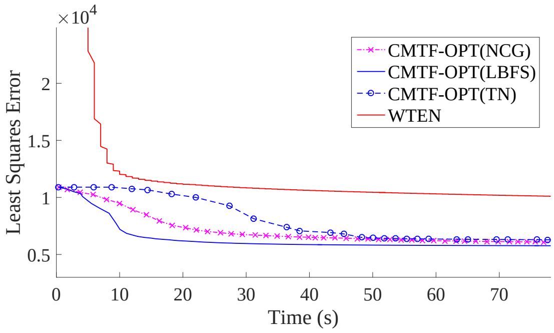


Fig. 5. Convergence of WTEN and CMTF-OPT on MovieLens training data. It is worth to note that the least squares error here is not the error rate that we evaluate in the experiment for final performance comparisons. Since the data is imbalanced, we use AUC as the comparison metric, as shown in Table 4.

6 Conclusion

We proposed a weighting strategy to provide Coupled Tensor Factorization method a capability to handle imbalanced data. Our work suggests three key learning insights. Firstly, our novel weighting strategy enables CTF to perform significantly better on the minority class prediction without jeopardizing the classification of the majority case. Secondly, our experiments demonstrate the impact of the additional matrix on CTF's performance over TF. This finding can serve as a foundation for a general purpose latent factor on imbalanced data learning. Thirdly, our factorization algorithm with weighting scheme can be easily extended to different imbalanced data sets and applications. Although this paper targets binary missing label estimation problem, the weighting strategy can be straightforwardly applied to multiple labels, such as integer ratings where the frequencies of integers are imbalanced. In the future, we are planning to scale up our idea using distributed computing environments.

References

[1] Australian bureau of statistics data sets, time series profile of local government areas http://www.abs.gov.au/websitedbs/censushome.nsf/home/datapacks

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

[3] Akbani, R., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In: Machine learning: ECML 2004. pp. 39–50. Springer (2004)

[4] Beutel, A., Talukdar, P.P., Kumar, A., Faloutsos, C., Papalexakis, E.E., Xing, E.P.: Flexifact: Scalable flexible factorization of coupled tensors on hadoop. In: SIAM International Conference on Data Mining (SDM). pp. 109–117 (2014)

[5] Bradford, J.P., Kunz, C., Kohavi, R., Brunk, C., Brodley, C.E.: Pruning decision trees with misclassification costs. In: Machine Learning: ECML-98, pp. 131–136. Springer (1998)

[6] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research pp. 321–357 (2002)

[7] Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. ACM SIGKDD Explorations Newsletter 6(1), 1–6 (2004)

[8] Elkan, C.: The foundations of cost-sensitive learning. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence-Volume 2 (2001)

[9] Fawcett, T.: An introduction to roc analysis. Pattern Recognition Letters 27(8), 861–874 (2006)

[10] Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. 5(4), 19:1–19:19 (Dec 2015)

[11] Harshman, R.A.: Foundations of the parafac procedure: Models and conditions for an" explanatory" multi-modal factor analysis (1970)

[12] He, H., Garcia, E., et al.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21(9), 1263–1284 (2009)

[13] Jo, T., Japkowicz, N.: Class imbalances versus small disjuncts. ACM SIGKDD Explorations Newsletter 6(1), 40–49 (2004)

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

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

[16] Ling, C.X., Yang, Q., Wang, J., Zhang, S.: Decision trees with minimal costs. In: Proceedings of the 21st International Conference on Machine Learning (2004)

[17] Liu, W., Chan, J., Bailey, J., Leckie, C., Ramamohanarao, K.: Mining labelled tensors by discovering both their common and discriminative subspaces. In: SIAM International Conference on Data Mining (SDM13). pp. 614–622 (2013)

[18] Liu, W., Kan, A., Chan, J., Bailey, J., Leckie, C., Pei, J., Kotagiri, R.: On compressing weighted time-evolving graphs. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM 2012). pp. 2319–2322 (2012)

[19] Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics 39(2) (2009)

[20] Mani, I., Zhang, I.: knn approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings of Workshop on Learning from Imbalanced Datasets (2003)

[21] Menon, A.K., Elkan, C.: Link prediction via matrix factorization. Machine Learning and Knowledge Discovery in Databases 6912, 437–452 (2011)

[22] Papalexakis, E.E., Faloutsos, C., Mitchell, T.M., Sidiropoulos, N.D.: Turbo-smt : Accelerating coupled sparse matrix-tensor factorizations by 200x. SIAM International Conference on Data Mining (SDM) (2014)

[23] Ristanoski, G., Liu, W., Bailey, J.: Discrimination aware classification for imbalanced datasets. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM 2013). pp. 1529–1532 (2013)

[24] Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). pp. 650–658 (2008)

[25] Wu, G., Chang, E.Y.: Kba: Kernel boundary alignment considering imbalanced data distribution. IEEE Transactions on Knowledge and Data Engineering (2005)

[26] Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering 18(1) (2006)