Mining Labelled Tensors by Discovering both their Common and Discriminative Subspaces

Wei Liu∗ Jeffrey Chan∗ James Bailey∗ Christopher Leckie∗
and Kotagiri Ramamohanarao∗

Abstract
Conventional non-negative tensor factorization (NTF) methods assume there is only one tensor that needs to be decomposed to low-rank factors. However, in practice data are usually generated from different time periods or by different class labels, which are represented by a sequence of multiple tensors associated with different labels. This raises the problem that when one needs to analyze and compare multiple tensors, existing NTF is unsuitable for discovering all potentially useful patterns: 1) if one factorizes each tensor separately, the common information shared by the tensors is lost in the factors, and 2) if one concatenates these tensors together and forms a larger tensor to factorize, the intrinsic discriminative subspaces that are unique to each tensor are not captured. The cause of such an issue is from the fact that conventional factorization methods handle data observations in an unsupervised way, which only considers features and not labels of the data.

To tackle this problem, in this paper we design a novel factorization algorithm called CDNTF (common and discriminative subspace non-negative tensor factorization), which takes both features and class labels into account in the factorization process. CDNTF uses a set of labelled tensors as input and computes both their common and discriminative subspaces simultaneously as output. We design an iterative algorithm that solves the common and discriminative subspace factorization problem with a proof of convergence. Experiment results on solving graph classification problems demonstrate the power and the effectiveness of the subspaces discovered by our method.

1 Introduction
In this paper we focus on the problem of non-negative tensor factorization (NTF) for a sequence of multiple tensors. Computing low-rank non-negative approximations of high dimensional data has become a major focus in data mining domain. The non-negativity property is desirable in many practical domains where non-zero data entries are usually positive, such as the adjacency matrices in graph mining domains and the images/videos analyzed in computer vision.

Many tensor factorization methods have been proposed in the past, such as Tucker decomposition [24], canonical polyadic decomposition [5] (CP, also known as CANDECOMP/PARAFAC [11]) and the NTF [25]. All of these methods and their later variants can be considered as higher-order generalizations of matrix factorizations. However, these existing methods are all restricted to decomposing a single instance of a tensor object in an unsupervised manner. This raises the question of what strategy should be used when dealing with multiple tensor objects which are associated with class labels. Given a tensor of $M$ dimensions, existing NTF methods decompose that tensor into $M$ low-rank matrix factors, each of which explains a compact basis of each dimension of the tensor. Two common approaches for using NTF to factorize a sequence of $m$ tensors are: (option 1) decompose each tensor separately (Figure 1a) – this approach generates a low-rank factor matrix for each mode of each tensor, and does not necessarily identify a potentially important “common factor” matrix that these tensors may share; or (option 2) concatenate all tensors to form one big tensor and then decompose it (Figure 1b) – in contrast to the first option, this strategy may discard all possible “discriminative factor” matrices in the concatenated dimension, and only produces factors that are a consensus of the original tensors. Although it is possible to treat these tensors as a data stream and use sliding windows to analyze them incrementally [22, 23], the actual decomposition on each element within each window is still limited to the above two options.

In this research we propose a novel strategy for an-
(a) Factorize each tensor independently by NTF, where matrix factors contain patterns that are unique to each tensor.

(b) Factorize the concatenated big tensor by NTF, where matrix factors only contain common patterns from all the tensors.

(c) Factorize tensors all together by our method CDNTF, where matrix factors in the first dimension contain both common and discriminative patterns of the tensors.

Figure 1: An example of a sequence of \( m \) tensors and three different approaches to factorizing them. “Tensor 1”, “Tensor 2”... represent data instances associated with different class labels. In the factor matrices, a subscript represents the index of dimensions and a superscript represents the index of tensors. In Fig (c) the sub-matrix \( U^* \) located in factor matrices of the first dimension denotes a common subspace shared by all tensors.

Factorizing multiple tensors, of which the two options above become special cases. We introduce the concepts of a **common subspace** and a **discriminative subspace** along a common dimension of all tensors. As demonstrated in Figure 1c, the common space (denoted by \( U^* \)) along the first dimension of all tensors occupies a fraction of the factor matrix, while the remaining fraction is preserved for each tensor independently so that any discriminative patterns that are unique to each tensor are also preserved. When the tensors are associated with class labels, the factorization of both common and discriminative subspaces is especially beneficial for classification tasks:

- The common subspaces are linear combinations of original features which are shared by data observations from all classes. The variance of data instances preserved in common subspaces are well kept for all classes. In this regards, the discovery of common subspaces is essentially a typical feature extraction (dimension reduction) of the original feature space for all data samples. The fact that common subspaces are shared by instances from all classes does not mean they are ineffective for learning classification models. Rather, the common subspaces can still be helpful for construction classification boundaries among data instances, since the (new) feature values that lie in these common subspaces can be very different among different classes.
- The discriminative subspaces are linear combinations of original features which are unique to each class label. Using a binary-class data set as an example, the variance of positive instances is only well kept in positive discriminative subspaces, while negative subspaces only preserve the variance of instances with negative class labels. Such class-contrasting properties of discriminative subspaces make them exceptionally suitable for learning classification boundaries among class labels.

More importantly, the common subspace factorization strategy provides a flexible choice on the sizes of common and discriminative factors, such that the preceding two options (shown in Figures 1a and 1b) become special cases of our proposed approach. When the common space is empty (i.e., when the size of \( U^* \) is zero) we obtain **option 1**, and when it is set to the full size of the factor matrix (instead of a fraction) we obtain **option 2**. Since the application of the common space to other...
dimensions is theoretically the same as applying it to one dimension, in this paper we focus on applying the common space strategy to one dimension. In summary, we make the following contributions in this paper:

1. We introduce the concepts of a common subspace and a discriminative subspace in factorizing a sequence of tensors, and formulate the problem of approximating low-rank representations of tensors as simultaneously optimizing the approximation of both the common and the discriminative subspaces;

2. We propose the CDNTF (common and discriminative subspace non-negative tensor factorization) algorithm that iteratively solves the above optimization problem, and provide a theoretical proof of convergence of the algorithm;

3. We perform empirical evaluations of CDNTF on solving graph classification problems, which demonstrates the power and effectiveness of our method.

The rest of the paper is structured as follows. We review related work in Section 2. In Section 3 we define the common and discriminative subspace factorization problem and introduce the CDNTF algorithm to solve it. The proof of convergence of the algorithm is deferred to the supplementary material. We explain how CDNTF can be applied to solve practical graph classification problems in Section 4. Details of our experimental evaluations are reported in Section 5. We conclude in Section 6 with directions for future work.

2 Related Work

Matrix and tensor factorization methods have become a major focus of research in recent years due to their effectiveness in solving many practical problems, including text mining [3], data stream mining [22], collaborative filtering [20], and link prediction [8] (to name a few). An extensive study on various existing types of matrix/tensor factorizations and their applications can be found in [13]. In this section we only review the methods that are closely related to our research.

2.1 Matrix Factorization For data observations in matrix form, a variety of factorization techniques have been developed that decompose matrices into low-rank factors. Examples of these techniques include principal component analysis (PCA), singular value decomposition (SVD), latent semantic analysis (LSA) [6], and non-negative matrix factorization (NMF) [19]. The non-negative property of factors decomposed by NMF has been used to solve various practical problems. For example, Gupta et al. [10] used NMF to tackle information retrieval problems by combining different sources of information into one querying process.

2.2 Tensor Factorization While the above approaches are suited to two dimensional matrix data, in more complex data sets with large numbers of dimensions, tensor factorizations are needed to find low-rank representations of the data. Factorization methods for tensors that are essentially higher order generalizations of those for matrices have been reported, such as Tucker decomposition [24], CP [5], and NTF [21]. As a multi-dimensional generalization of NMF, NTF [21] is more attractive than NMF not only because it considers more dimensions of information, but also because it usually allows for a unique decomposition of a data set into factors under mild conditions, which are usually satisfied by real data [12]. An example of this point is the work of Morup et al. [17] which developed a method for NTF while imposing sparseness constraints to enhance uniqueness. Sun et al. [22,23] proposed methods for incremental tensor analysis, which are aimed at solving the decomposition problem when there is a stream of many tensors. However, their methods factorized each tensor separately and only detected patterns that were unique to each tensor, which could not tell the common patterns of the low-rank factors among different tensors.

2.3 Coupled Matrix and Tensor Factorization It is proposed in Acar et al. [1] to decompose a coupled pair of a matrix and a tensor, where the matrix and the tensor only share one identical mode. However, the authors only consider the case when the decomposed factors are all the same in the shared mode, and did not address how to discover discriminant factors from the shared identical mode between coupled matrices or tensors. Lin et al. [16] proposed the MetaFac method with applications in community discovery. MetaFac uses several tensors to present relations of different facets in a hypergraph, where the tensors only share some of their modes. However, when it factorizes the tensors on the shared modes, only common factors are produced. Hence the question of how to discover the discriminative (e.g., different) factors that lie in the same shared modes among different tensors is still unsolved. Since these discriminant factors contain the information that distinguishes the original tensors, it is
important to unveil them so that the original tensors can be better and more easily differentiated (e.g., for classification applications).

The major difference between our research and all the above existing methods is that, we simultaneously decompose a set of tensors into both common and discriminative factors (on any mode of the tensors). To the best of our knowledge, our research proposes the first tensor decomposition method that discovers multiple tensors’ common and discriminative patterns by using both their features and class labels in the factorization process.

3 Tensor Factorization

Tensors are multidimensional arrays. The dimensionality of a tensor is also called the tensor’s mode. We denote tensors with 3 or more dimensions by letters of calligraphic font (e.g., \( \mathcal{X} \)), denote matrices (tensors with 2 dimensions) by boldface lowercase letters (e.g., \( \mathbf{U} \)), and denote vectors (tensors with 1 dimension) by boldface lower letters (e.g., \( \mathbf{u} \)). Table 1 lists the notation we use in the paper. We give the definition for a standard non-negative tensor factorization as follows:

**Definition 1. (NTF):** Given a M-dimension non-negative tensor \( \mathcal{X} \in \mathbb{R}^{n_1 \times \ldots \times n_M} \) and the desired low rank \( r \), NTF factorizes \( \mathcal{X} \) into M non-negative matrix factors \( \mathbf{U}_d \in \mathbb{R}^{n_d \times r}, (d = 1, 2, 3, \ldots, M) \), such that it minimizes the factorization error \( \| \mathcal{X} - \sum_{d=1}^{M} u_d^1 \otimes u_d^2 \otimes \ldots \otimes u_d^M \| \), where \( u_d^j \) represents the \( j \)th column of \( \mathbf{U}_d \), and \( \otimes \) represents outer products.

In other words, NTF solves the minimization problem: \( \min \{ f(\mathbf{U}_d) = \min \| \mathcal{X} - \mathbf{U}_1 \mathbf{U}_2 \ldots \mathbf{U}_M \| \} \). Note that for brevity, we use \( \mathbf{U}_1 \mathbf{U}_2 \ldots \mathbf{U}_M \) or \( \mathbf{U}_d |_{d=1}^{M} \) (or simply \( \mathbf{U} \)) to represent the operation \( \sum_{d=1}^{M} u_d^1 \otimes u_d^2 \otimes \ldots \otimes u_d^M \) in the rest of the paper.

### 3.1 Common and Discriminative Subspace Factorization

Without loss of generality, we define and solve the problem of learning a common subspace from multiple tensors by using the scenario of two tensors (e.g., \( m = 2 \) in the example of Figure 1). This scenario corresponds to the case of binary classes, where each tensor contains instances from a class label. We omit the lengthy derivations for \( m > 2 \) scenarios due to their close theoretical similarity to the \( m = 2 \) scenario. The common subspace learning problem for two tensors can be defined as follows:

**Definition 2. (Common and Discriminative Subspace NTF):** Given two non-negative tensors \( \mathcal{X}_1 \) and \( \mathcal{X}_2 \), common subspace factorization decomposes each of them as the product of \( M + 1 \) non-negative matrices so that the factorization errors \( \| \mathcal{X}_1 - [W]V U_1 U_2 \ldots U_M \| \) and \( \| \mathcal{X}_2 - [W]S K_1 K_2 \ldots K_M \| \) are both minimized.

In this definition, we use “[\( W \)\( V \)]” and “[\( W \)\( S \)]” to represent the first matrix factor of each tensor, where \( W \) is the common subspace, and \( V \) and \( S \) are the remaining discriminative subspaces of the two tensors. In the example of Figure 1c, \( W \) is equivalent to \( U^* \), and \( V \) and \( S \) are equivalent to \( U_1^2 \) and \( U_2^2 \). Since the common subspace is located in the first dimension of the tensors, \( W \) is of size \( n_1 \times k \), while \( V \) and \( S \) are of size \( n_1 \times (r - k) \).

By introducing the Frobenius norm and using \( \mathbf{U} \) and \( \mathbf{K} \) to respectively denote the product of factor matrices \( \mathbf{U}_d \) and \( \mathbf{K}_d \) (\( 2 \leq d \leq M \)), we know from Definition 2 that common subspace factorization on two tensors is equivalent to the minimizations of the following equations simultaneously:

### 3.2 The Proposed Algorithm: CDNTF

We solve the objective in Eq. 3.3 by forming a positive-preserving
Algorithm 1 CDNTF: algorithm for solving the common and discriminative subspace tensor factorization problem.

**Input:** original tensors X1 and X2, r, k (see Table 1 for descriptions of r and k), and an error threshold ε.

**Output:** common and discriminative matrix factors.

1: Randomly initialize W, V, U_d, S, and K_d;
2: Initialize the change of objective J: ΔJ ← +Infinity;
3: while (ΔJ > ε) and (not exceed maxNumIters) do
4:    Update W, V, U_d, S, and K_d by Eq. 3.4 to 3.8;
5:    Normalize the columns of W, V and S to unit length;
6:    Update J and compute its change ΔJ by Eq. 3.3;
7: end while
8: return W, V, U_d, S, and K_d.

is a special case of CDNTF when k = 0.

4 Application of CDNTF in Graph Mining

In this section we briefly demonstrate how the theory of common subspace factorization can address graph classification problem, which is an important problem in graph mining. One reason for selecting graph data to validate CDNTF is that when graphs are complex and dynamic, there is always more information than just their adjacency matrices that need to be considered, such as the labels of vertices\(^2\) and the weights of edges. This diverse information on graph data puts the use of tensors at an advantage.

We apply CDNTF to classification problems by making use of the new data points that lie in the new low-dimension feature spaces defined by the low-rank factor matrices W, V and S. Given a set of graphs, each of which is associated with a class, the task is to predict the class of a new graph. If there are two classes in total and we construct one tensor for each class, then we obtain two tensors, X1 and X2, having 4 modes: edges × vertexlabels × edgeweights × observations(graph instances), where the first to third mode respectively consists of all unique edges (vertex pairs), unique vertex labels, and unique edge weights that are found in a data set. So an entry of these two tensors tells that, for a certain graph (i.e., the fourth mode), how many edges (i.e., the first mode) are connected to some vertex labels (i.e., the second mode) and have certain edge weights (i.e., the third mode). One can think of the 4-mode tensors as standard features × observations data sheets where the features are represented by 3-mode tensors.

Assume it is the first edge mode on which we want

\(^2\)The proof of convergence of the iterations in Alg. 1 is provided in a supplementary material at http://rp-wv.cs.usyd.edu.au/~weiliu/Webpage/SDM2013_Supplementary.pdf

\(^3\)We note that the “labels of vertices”, which is a property (i.e., features) of a graph, should not be confused with “labels of graphs”, which is the class of the graph.
to discover the common and discriminative subspaces, then \textsc{CDNTF} produces the matrix factors \([W|V]\) and \([W|S]\) on the first mode of the two tensors respectively. The columns of matrices \(W, V\) and \(S\) are non-negative linear combinations of the original edges, which define the new low-rank feature spaces. Given original data points \(D_{\text{original}}\) (i.e., either \(X_1\) or \(X_2\)), we obtain new data points \(D_{\text{new}}\) which lie in such new low-rank feature spaces by two steps: (1) obtain new coordinates \(D_W\), \(D_V\) and \(D_S\) in the new spaces (defined by \(W, V\) and \(S\) respectively) by solving the following two linear equations (using Gaussian elimination [9]):

\[
[W|V][D_W|D_V] = D_{\text{original}},
\]

and then (2) concatenate the new coordinates together to form new data points: \(D_{\text{new}} = [D_W|D_V|D_S]\). In other words, the new data points are the original data \((X_1\) and \(X_2)\) that are “projected” into the new space defined by \(W, V\) and \(S\). When \(D_{\text{new}}\) is of multi-mode (e.g., solved from high-mode tensors), one can matricize (e.g., unfold) \(D_{\text{new}}\) into a matrix by fixing its observations (instances) mode and varying other modes, so that each graph instance becomes a vector and can be learned by any standard classifiers. Intuitively, these new coordinates \(D_{\text{new}}\) makes full use of the power of \textsc{CDNTF}, since they are characterized by both the common space \(W\) and the discriminative spaces \(V\) and \(S\).

Note that the generation of new data points includes the scenario of standard NTF (when the size of \(W\) is zero), and also includes that of CP tensor decomposition (when there is no non-negativity constraint). It also includes methods of graph classification by frequent subgraph extraction [14] [15], since those methods use new data points that lie in the feature space formed by a vector of subgraphs. In this case, a subgraph is essentially a (binary) linear combination of the original vertices/edges, which is similar to the columns of \(W, V\) and \(S\). In this regard, the advantage of \textsc{CDNTF} in graph classification is that the new data points it produces are superior abstraction of the original graphs, which do not lose common subspace information in the feature space (which standard NTF does). Moreover, it does not restrict the vertices/edges in each new feature space (i.e., each linear combination) to be fully connected in original graphs (which subgraph extraction methods do).

5 Experiments and Analysis

Besides the theoretical convergence guarantee of \textsc{CDNTF}, in this section we would like to show that our method also works well in practice. We implement \textsc{CDNTF} and NTF in Matlab by using the Tensor Toolbox [2]. This toolbox also contains an implementation of CP tensor decomposition, which we use in the evaluation. All experiment results presented in this section are from 5-fold cross validation with 10 repeated runs.

Table 2: Statistics of chemical compound data sets. “pos%” represents the proportion of compounds that are associated with positive classes.

<table>
<thead>
<tr>
<th>Name</th>
<th>#graphs</th>
<th>pos%</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AID83</td>
<td>27784</td>
<td>8.2</td>
<td>Breast Cancer</td>
</tr>
<tr>
<td>AID81</td>
<td>40700</td>
<td>5.9</td>
<td>Colon Cancer</td>
</tr>
<tr>
<td>AID123</td>
<td>40152</td>
<td>7.8</td>
<td>Leukemia</td>
</tr>
<tr>
<td>AID1</td>
<td>40460</td>
<td>5.1</td>
<td>Lung Cancer</td>
</tr>
<tr>
<td>AID33</td>
<td>40209</td>
<td>4.1</td>
<td>Melanoma</td>
</tr>
<tr>
<td>AID47</td>
<td>40447</td>
<td>5.0</td>
<td>Nerve Cancer</td>
</tr>
<tr>
<td>AID109</td>
<td>40691</td>
<td>5.1</td>
<td>Ovarian Cancer</td>
</tr>
<tr>
<td>AID41</td>
<td>27585</td>
<td>5.7</td>
<td>Prostate Cancer</td>
</tr>
<tr>
<td>AID45</td>
<td>40164</td>
<td>4.9</td>
<td>Renal Cancer</td>
</tr>
<tr>
<td>AID1481</td>
<td>217968</td>
<td>12.6</td>
<td>ATPase Inhibition</td>
</tr>
<tr>
<td>AID1416</td>
<td>217968</td>
<td>19.8</td>
<td>PERK Inhibition</td>
</tr>
<tr>
<td>AID1449</td>
<td>217968</td>
<td>11.4</td>
<td>Janus Kinase</td>
</tr>
<tr>
<td>AID1531</td>
<td>289475</td>
<td>11.7</td>
<td>Mek Inhibitors</td>
</tr>
<tr>
<td>PTC-M</td>
<td>336</td>
<td>41.0</td>
<td>Mice Toxic</td>
</tr>
<tr>
<td>PTC-R</td>
<td>349</td>
<td>38.4</td>
<td>Rats Toxic</td>
</tr>
</tbody>
</table>

5.1 Chemical Compound Data Sets We apply \textsc{CDNTF} to the graph classification problem on chemical compound data sets, where each compound is treated as a graph. Two types of chemical compound are used: (1) Bioassays of anti-cancer activity and kinase inhibition (AID)\(^4\); the task is to predict whether a compound is positive or negative in anti-cancer activities or in kinase inhibition activities. The original data sets contain a large number of compounds (shown in Table 2). We randomly sample 1000 compounds from each data set for evaluation. (2) Toxicology prediction (PTC)\(^5\): the task is to predict the carcinogenicity of compounds on mice and rats. Each chemical compound is associated with a carcinogenicity class from \{CE, SE, P, E, EE, IS, NE, N\}. Following the settings of [15], we use \{CE, SE, P\} as positive classes, \{NE, N\} as negative ones, and discard other neutral classes. Details of these chemical compound data sets are reported in Table 2. For the parameter settings in Alg. 1, we set the convergence threshold \(\epsilon\) to \(10^{-4}\), and vary the value of rank \(r\) to validate the performance of our method.

5.2 Graph Classification Similar to the discussion in Section 4, in each data set we construct two tensors, one for each class, where all unique types of vertices found in a data set are converted to the labels of vertices, and


http://www.predictive-toxicology.org/ptc/
the lengths of bonds between atoms are weights of the edges. We partition (discretize) all edge weights into 100 bins (with equal width) in each data set. So an entry of a tensor tells that for a certain compound, how many edges (bonds) connect certain types of atoms and have certain length (weight). Due to the different nature and properties of each data set, we find the best value of \( k \) (the size of the common spaces) empirically from the best fit within the training data of cross validations.

We note that the imbalanced nature of classes in compound data sets only influences the cardinality of the “instance” dimension (the last mode in the example stated in Section 4) of the two constructed tensors, and does not affect the low-rank approximation of other “feature” dimensions. Therefore the imbalance does not influence the formulations of the factor matrices \( W, V \) and \( S \).

We compare the accuracy of classification on data points lying in the new feature space produced by NTF, CDNTF and CP decomposition, where we vary the settings of low ranks (\( r \)) from 5 to 20. In NTF, all data instances are factorized together, so it only discovers common subspaces of both classes. In CP, tensors belonging to different classes are factorized separately, hence it only finds discriminative subspaces of the classes. We also include the MetaFac method [16] in our evaluations, which produces common factors on the common mode (called “facet” in MetaFac) of tensors.

We use two types of classifiers to learn from the new data points, a linear classifier – logistic regression, and a non-linear classifier – support vector machines (SVM) with quadratic kernels, and perform 5-fold cross validation with 10 repeated runs on each data set. As can be observed from Figure 2, the data points that lie in the low-rank spaces produced by CDNTF are always much easier to be distinguished (classified), by both logistic regression and SVMs, than the data points that lie in the spaces produced by NTF, CP and MetaFac.
Because of the class imbalance, we use the area under the ROC curve (AUC) as the evaluation metric, which is more appropriate than using accuracy as the metric. Due to page limits, in Figure 2 we only present the results of three data sets (i.e., breast cancer, colon cancer and melanoma), to which the results from other data sets make the same conclusions.

To confirm the significance of the superiority of CDNTF, we perform Friedman tests on the sequences of AUC values across all data sets, where p-values that are lower than 0.05 reject the hypothesis with 95% confidence that the classifiers in the comparison are not statistically different. The Friedman test is reported as the most appropriate method for validating multiple classifiers among multiple data sets [7], so we take the Friedman tests within the same type of classifiers, where CDNTF is the base method against which the other methods are compared. In Tables 3 and 4 we report the performance of the classifiers on different factorization methods when the rank is 20, where the values of means and standard deviations are obtained from the repeated runs of cross validation. In each data set the AUC value of the best performing method is put in boldface font. To show the diversity of the data sets, we also present the best k values which are optimized from the training set of cross validation. From the low p-values shown in the bottom of Table 3 and 4, it is easy to see that the low-rank spaces produced by CDNTF are significantly better than the other corresponding methods in distinguishing the two class labels on each data set.

Since our method is evaluated on the task of graph classification, we include frequent subgraph extraction method in our comparisons. In frequent subgraph extraction we use a support threshold of 0.1. Since closed graphs avoid subgraph correlation problems [4], we also include the closed-subgraph extraction method in the evaluations. The performance of these subgraph extraction methods compared with CDNTF is reported in Table 5 and 6. The low p-values shown in the bottom of these two tables illustrate the advantage of our method in discriminating the class labels in the low-rank spaces.

### 6 Conclusions and Future Work

In this research we focus on the problem of factorizing a sequence of labelled tensors. We address the challenge of how one should decompose tensors such that, by taking into account of class labels, both common and discriminative patterns among different tensors can be interpreted simultaneously from the factor matrices. We formulate this problem into the task of learning common and discriminative subspaces from overall data instances, and solve it by the proposed CDNTF algorithm. We provide a theoretical proof for the convergence of CDNTF, and thus the optimality of the output of the algorithm is guaranteed. To the best of our knowledge, CDNTF is the first tensor factorization method that discovers both common and discriminative patterns simultaneously from multiple labelled tensors.

CDNTF is a highly generic method that can potentially be applied to many practical data mining problems. We have applied CDNTF to the problem of graph classifications using chemical compound data sets.

---

We used the implementation provided by gSpan [26], which is the underlying basis of many recent methods of graph classification [14, 15, 18].
Table 6: Comparisons of CDNTF and frequent subgraph extraction methods using logistic regression.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Freq. subg.</th>
<th>AUC from LogReg</th>
<th>Closed subg.</th>
<th>CDNTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>AID83</td>
<td>0.65±0.002</td>
<td>0.579±0.002</td>
<td>0.662±0.005</td>
<td></td>
</tr>
<tr>
<td>AID81</td>
<td>0.72±0.003</td>
<td>0.67±0.002</td>
<td>0.763±0.003</td>
<td></td>
</tr>
<tr>
<td>AID123</td>
<td>0.659±0.003</td>
<td>0.61±0.003</td>
<td>0.755±0.004</td>
<td></td>
</tr>
<tr>
<td>AID1</td>
<td>0.753±0.002</td>
<td>0.732±0.004</td>
<td>0.821±0.004</td>
<td></td>
</tr>
<tr>
<td>AID33</td>
<td>0.694±0.002</td>
<td>0.701±0.004</td>
<td>0.729±0.006</td>
<td></td>
</tr>
<tr>
<td>AID47</td>
<td>0.740±0.004</td>
<td>0.632±0.004</td>
<td>0.743±0.004</td>
<td></td>
</tr>
<tr>
<td>AID109</td>
<td>0.728±0.005</td>
<td>0.731±0.006</td>
<td>0.781±0.004</td>
<td></td>
</tr>
<tr>
<td>AID41</td>
<td>0.766±0.007</td>
<td>0.653±0.004</td>
<td>0.789±0.004</td>
<td></td>
</tr>
<tr>
<td>AID145</td>
<td>0.784±0.004</td>
<td>0.718±0.008</td>
<td>0.812±0.006</td>
<td></td>
</tr>
<tr>
<td>AID1481</td>
<td>0.721±0.002</td>
<td>0.747±0.002</td>
<td>0.781±0.006</td>
<td></td>
</tr>
<tr>
<td>AID1416</td>
<td>0.796±0.005</td>
<td>0.807±0.003</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AID1446</td>
<td>0.838±0.002</td>
<td>0.808±0.004</td>
<td>0.899±0.003</td>
<td></td>
</tr>
<tr>
<td>AID1531</td>
<td>0.809±0.004</td>
<td>0.831±0.004</td>
<td>0.871±0.003</td>
<td></td>
</tr>
<tr>
<td>PTC-M</td>
<td>0.574±0.013</td>
<td>0.579±0.012</td>
<td>0.601±0.009</td>
<td></td>
</tr>
<tr>
<td>PTC-R</td>
<td>0.663±0.011</td>
<td>0.624±0.010</td>
<td>0.669±0.012</td>
<td></td>
</tr>
<tr>
<td>Pred. test</td>
<td>✓ 0.0001</td>
<td>✓ 0.0001</td>
<td>Base</td>
<td></td>
</tr>
</tbody>
</table>

Empirical results demonstrate the superiority of the subspaces discovered by CDNTF over other alternative methods, including those that use only common subspaces and those that use only discriminative subspaces.

In future, we will investigate the use of CDNTF in sparse tensor factorization problems, where each subspace is from a subset of original features.

References


