A Generalized Hierarchical Multi-Latent Space Model for Heterogeneous Learning

Pei Yang, *Member, IEEE,* Hasan Davulcu, *Member, IEEE,* Yada Zhu, *Member, IEEE,* and Jingrui He, *Member, IEEE,*

Abstract—In many real world applications such as image annotation, gene function prediction, and insider threat detection, the data collected from heterogeneous sources often exhibit multiple types of heterogeneity, such as task heterogeneity, view heterogeneity, and label heterogeneity. To address this problem, we propose a Hierarchical Multi-Latent Space (HiMLS) learning framework to jointly model the triple types of heterogeneity. The basic idea is to learn a hierarchical multi-latent space by which we can simultaneously leverage the task relatedness, view consistency and the label correlations to improve the learning performance. We first propose a multi-latent space approach to model the complex heterogeneity, which is then used as a building block to stack up a multi-layer structure in order to learn the hierarchical multi-latent space. In such a way, we can gradually learn the more abstract concepts in the higher level. We present two instantiated models of the generalized framework using different divergence measures. The two-phase learning algorithms are used to train the multi-layer models. We drive the multiplicative update rules for pre-training and fine-tuning in each model, and prove the convergence and correctness of the update methods. The effectiveness of the proposed approach is verified on various data sets.

Index Terms—Heterogeneous learning, multi-task learning, multi-view learning, multi-label learning, matrix tri-factorization.

1 Introduction

In the era of big data, a large amount of information collected from heterogeneous sources are correlated with each other. It is of great importance to mine such hidden correlations in the presence of multiple types of heterogeneity for many real world applications, such as web news classification, gene function prediction, insider threat detection, image annotation, etc. In this paper, we focus on triple types of heterogeneity, i.e., task heterogeneity, view heterogeneity, and label heterogeneity. For example, for the satellite image analysis problems, task heterogeneity refers to the images collected from different satellites following from different distributions; view heterogeneity refers to various types of features such as color histogram, edge distribution histogram, and bag of visual words; label heterogeneity refers to the multiple tags associated with each image.

The major challenge for learning with the triple types of heterogeneity is how to effectively mine the hidden correlations among the heterogeneous data. Such correlations should reflect the key assumptions underlying each type of heterogeneity, including the task relatedness assumption [7], the view consistency assumption [16], as well as the label correlation assumption [34]. To the best of our knowledge, we are the first to jointly model the triple heterogeneity.

To tackle this problem, we propose a Hierarchical Multi-Latent Space (HiMLS) framework for heterogeneous learning. The goal is to maximally leverage the rich correlations among heterogeneous data to improve the performance. To this end, we first present a multi-latent space model, which characterizes task relatedness, view consistency, and label correlation in a principled framework. It is formulated as a regularized non-negative matrix tri-factorization problem, aiming to simultaneously minimize the reconstruction loss on the instance-feature data and the classification loss on the instance-label data, while maximizing the similarity among the co-latent spaces. Furthermore, the proposed multi-latent space model is used as a building block to establish a multi-layer structure. It aims to build a hierarchical multi-latent space to gradually learn the more abstract concepts in the higher layer. The proposed HiMLS approach is motivated from two streams of work in machine learning. One is multi-way clustering (or co-clustering) [3] which improves the quality of clustering by intertwining both row and column information that are inter-related. Another is multi-layer models [19] which obtains better data representations by automatically extracting the hierarchical concepts from data. Our multi-latent space model employs multi-way clustering on the instances, features, and labels to capture the correlations among the heterogeneous data, while the hierarchical multi-latent space model takes advantage of multi-layer structure to learn the hierarchical concepts from data. Both of them help extract the rich correlations among heterogeneous data, leading to better performance.

Based on this generalized framework, we present two instantiated models using different distance metrics, i.e., least squares loss function and the generalized Kullback-Leibler divergence. For each model, we develop an iterative updating algorithm to solve the optimization problem. The proposed algorithms consist of two phases. First, each layer is pre-trained in a greedy layer-wise way. Then, it fine-tunes the weights of all the layer to reduce the total reconstruction loss and the classification loss. It is worth noting that the proposed approach is a generalized framework to learn from complex heterogeneity, which subsumes some popular methods on learning from a single heterogeneity.

• *P. Yang, H. Davulcu, and J. He are with Arizona State University, Tempe, AZ 85281. E-mail: cs.pyang@gmail.com, HasanDavulcu@asu.edu, jingrui.he@gmail.com. Y. Zhu is with IBM Research, Yorktown Heights, NY 10598. E-mail: yzhu@us.ibm.com*

Manuscript received December 22, 2015; revised July 26, 2016.

The main contributions of this paper can be summarized as:

- A novel learning problem which simultaneously models triple types of heterogeneity;
- A generalized framework to learn the hierarchical multi-latent space from complex heterogeneity;
- Two alternative models and the corresponding optimization algorithms;
- Generalization of some previous work on learning from single heterogeneity;
- Experimental results on various data sets showing the effectiveness of the proposed approach.

The rest of the paper is organized as follows. After a review of the related work in Section 2, we present the proposed generalized framework in Section 3, and two alternative models and their corresponding optimization algorithms in Section 4 and 5, respectively. Some case studies are discussed in Section 6. Section 7 shows the experimental results. Finally, we conclude the paper in Section 8.

2 Related Work

Since we make use of matrix factorization techniques to model the complex heterogeneity, we review the related work on both heterogeneous learning and non-negative matrix factorization.

2.1 Heterogeneous Learning

Heterogeneous learning aims to leverage different types of heterogeneity, such as task heterogeneity, view heterogeneity, and label heterogeneity, to improve the learning performance. Most of the previous work were focused on modeling a single or dual types of heterogeneity.

In multi-task learning, the goal is to leverage the small amount of labeled data from multiple related tasks to improve the learner for each task. Among others, alternating structure optimization [1] decomposed the model into the task-specific and task-shared feature mapping; multi-task feature learning [2] assumed that multiple related tasks share a low-dimensional representation; clustered multi-task learning [47] assumed that multiple tasks follow a clustered structure. Some recent multi-task learning methods dealt with irrelevant tasks by assuming that the model can be decomposed into a shared feature structure that captures task relatedness, and a group-sparse structure that detects outliers [17].

In multi-view learning, the features from multiple sources form natural views. The goal is to leverage the complementary information among different views to improve the performance. Co-Training [4] is one of the earliest algorithms for multi-view learning. More recent work includes: SVM-2K [16] which combined KCCA with SVM in an optimization framework; the information-theoretic framework for multi-view learning [30]; the CoMR method [29] based on a data-dependent Reproducing Kernel Hilbert Space (RKHS); the large-margin framework for multi-view data based on a latent space Markov network [8]; the convex multi-view subspace learning method MSL [36] which enforced conditional independence among the multiple views while reducing dimensionality, etc.

In multi-label learning, each instance is associated with a set of labels [34], [46]. The key issue is how to exploit the correlations or dependencies among multiple labels. To name a few, ML-kNN [45] converted the multi-label learning into a number of independent binary classification problems; Rank-SVM [15] solved the label ranking problem under the large margin framework; LEAD [44] employed Bayesian network to encode the conditional dependencies of the labels; LS-ML [22] assumed that a common subspace is shared among multiple labels; HG [31] constructed a hypergraph to exploit the correlation information among different labels; LEML [41] learned the latent label space under a generic empirical risk minimization framework with trace-norm regularization. In addition, MLLOC [21] assumed that the label correlation may be shared by a subset of instances only rather than all the instances; the boosting based method MAHR [20] aimed to discover the label relationship by using a hypothesis reuse mechanism; the transductive approach TRAM [23] leveraged the information from unlabeled data to estimate the optimal label concept compositions.

More recently, researchers begin to study problems with dual types of heterogeneity. For problems with both task and view heterogeneity, a variety of techniques have been proposed to model task relatedness in the presence of multiple views, e.g., the transductive method $IteM^2$ [18], the inductive method regMVMT [43], the bayesian method NOBLE [37], and the graph-based method M^2LID [39]. For the problems with both label and view heterogeneity, the L^2F method proposed in [40] modeled both the view consistency and the label correlations in a graph-based framework. For the more complex setting with all three types of heterogeneity, these techniques cannot be readily applied without disregarding the useful information from a certain type of heterogeneity, except for our recent work [38] on modeling the triple heterogeneity. This paper extends [38] substantially by providing the generalized learning framework, the alternative optimization algorithms, and the theoretical analysis regarding the optimal solutions, as well as the comprehensive empirical evaluations.

2.2 Non-Negative Matrix Factorization

Non-negative matrix factorization (NMF) [24] aims to extract data-dependent non-negative basis functions, which has been given much attention due to its part-based and easy interpretable representation. Non-negative matrix factorization [25] has been widely used in data mining, biomedical, chemometrics, signal processing, computer vision, neuroscience, graph analysis, etc [10]. Incorporating extra constraints such as sparseness [28], smoothness [5], or orthogonality [14] was shown to improve the decomposition and provide the better representation. Various extensions and variations of NMF have been proposed, such as Semi-NMF [12], Convex-NMF [12], multi-layer NMF [10], [33], weighed NMF [35], Tri-NMF [14], etc.

NMF has connections to many other techniques in data mining. For example, under some mild conditions, NMF with the least squares loss function is equivalent to a relaxed K-means clustering [11], while NMF with the generalized Kullback-Leibler (KL) divergence loss function is equivalent to probabilistic latent semantic indexing [13].

3 The Proposed Generalized Framework for Heterogeneous Learning

We first present the multi-latent space framework to model the complex heterogeneity, which is then used as a building block to stack up a multi-layer structure in order to learn the hierarchical multi-latent space.

3.1 Notations and Problem Statements

Suppose we are given the multi-label data with multiple views in different tasks. Let T be the number of tasks, V the number of views, m the number of labels. Each instance is described from V views, and associated with multiple labels. For the i^{th} task and j^{th} view, denote the number of instances and features by n_i and d_j, respectively. Let $\tilde{X}_{ij}=\begin{bmatrix} X_{ij} \\ X_{ij}^u \end{bmatrix} \in \mathcal{R}^{n_i \times d_j}$ be the instance-feature matrix for the i^{th} task and j^{th} view, where X_{ij} is the training data and X_{ij}^u is the test data. Let $\tilde{Y}_i = \begin{bmatrix} Y_i \\ Y_i^u \end{bmatrix} \in \mathcal{R}^{n_i \times m}$ be the instance-label matrix for the i^{th} task, where Y_i and Y_i^u are for training and test data respectively. The instance-label matrix can be either a binary or a real matrix, such as the user-item matrix of either preference or rating scores in a recommender system. The goal is to build a model to predict the instance-label matrix for the test data by leveraging the rich information among heterogeneous data.

Some math symbols used in this paper are introduced as follows. For two matrices X and Y, let $X \odot Y$, $X \otimes Y$, and $\frac{X}{Y}$ be the Hadamard product (or entrywise product), Kronecker product, and Hadamard division, respectively. Let $x = vec(X)$ be the matrix vectorization of X into a vector x.

3.2 Multi-Latent Space Learning

We propose a multi-latent space learning framework to jointly model the task relatedness, view consistency, and label correlations in a principled way.

Motivated by the success of multi-way clustering [3] in leveraging the inter-correlations among data to improve clustering quality, we do multi-way clustering on heterogeneous data to learn the multi-latent space. It simultaneously clusters instances, features and labels into the corresponding clusters. Let $\tilde{R}_i = \begin{bmatrix} R_i \\ R_i^u \end{bmatrix} \in \mathcal{R}^{n_i \times p}$ be the instance encoding matrix where p is the dimensionality of instance latent space, R_i and R_i^u are for training and test data, respectively. Let $C_j \in \mathcal{R}^{d_j \times q}$ be the feature encoding matrix, $C_Y \in \mathcal{R}^{m \times q}$ the label encoding matrix where q is the dimensionality of feature (or label) latent space. Each row in $\tilde{R}_i$ (or C_j, C_Y) represents the coeffecients of the instance (or feature, label) associated with the instance (or feature, label) clusters. Denote $M_{ij} \in \mathcal{R}^{p \times q}, M_{iY} \in \mathcal{R}^{p \times q}$ as the co-latent space matrices. We try to reconstruct the instance-feature matrix and instance-label matrix by letting $\tilde{X}_{ij} \approx \tilde{R}_i M_{ij} C_j^T$ and $Y_i \approx R_i M_{iY} C_Y^T$ respectively, where $1 \leq i \leq T$ and $1 \leq j \leq V$. Note that M_{ij} models the correlations between instance clusters and feature clusters, while M_{iY} models the correlations between instance clusters and label clusters.

The multi-latent space model is formulated as a regularized non-negative matrix triple factorization problem, which simultaneously decomposes the instance-feature and instance-label matrices, while enforcing the task relatedness, view consistency, and label correlations on the data. The objective is to simultaneously minimize the reconstruction loss on the instance-feature data (1st term) and the classification loss on the instance-label data (2nd term), while maximizing the similarity among the co-latent spaces (3rd term):

$$\begin{aligned} \min_{\{R,M,C\}>0} & \sum_{i=1}^{T}\sum_{j=1}^{V} \mathcal{L}\left(\tilde{X}_{ij}; \tilde{R}_i M_{ij} C_j^T\right) \\ & + \alpha \sum_{i=1}^{T} \mathcal{L}\left(Y_i; R_i M_{iY} C_Y^T\right) + \beta \sum_{i=1}^{T}\sum_{j=1}^{V} \mathcal{L}\left(M_{ij}; M_{iY}\right) \end{aligned} \tag{1}$$

where $\mathcal{L}(X;Y)$ is the distance metric between X and Y. α and β are the non-negative parameters. The non-negative constraints allow for the multi-way clustering interpretation.

The multi-latent space model can be interpreted from the perspective of constrained multi-way clustering. By constraining the multi-way clustering procedures, we model the **task relatedness** by requiring the features across different tasks to share the same feature clustering coefficients, enhance the **view consistency** by requiring the instances to share the same instance clustering coefficients across different views, characterize the **label correlations** by requiring the labels to share the same label clustering coefficients across different tasks. Figure 1(a) shows an illustrative example about the proposed multi-latent space model. Specifically, the multi-latent space model encodes multiple types of correlations among the heterogeneous data as follows:

Task relatedness: For the j^{th} view, the decompositions of the instance-feature data X_{ij} $(1 \leq i \leq T)$ in different tasks share the same feature encoding matrix C_j.

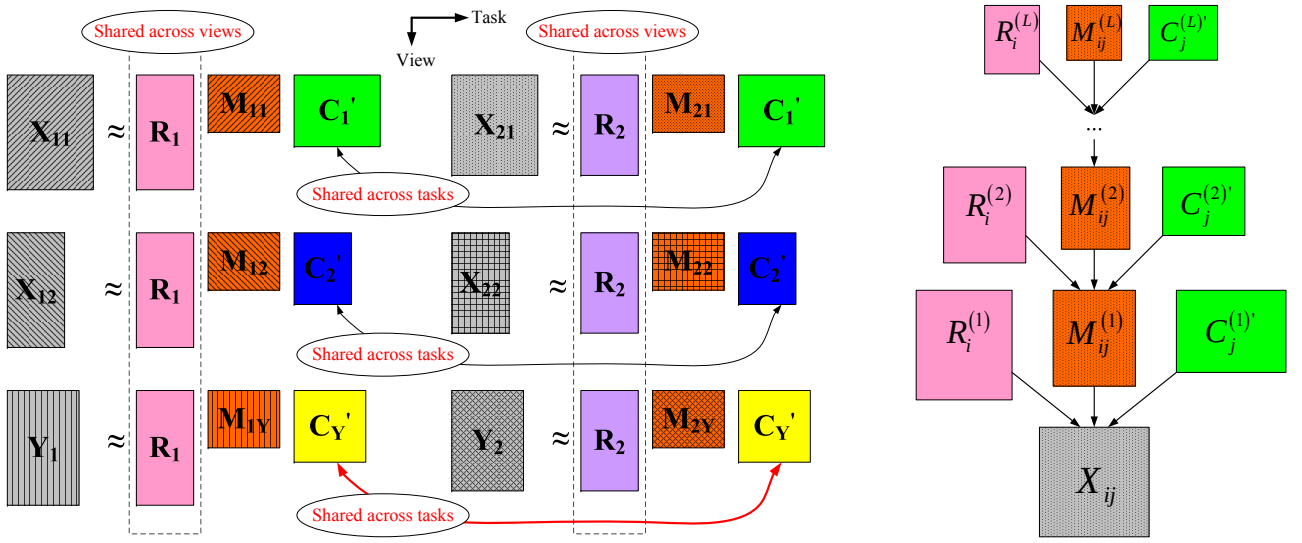
Label correlation: The labels share the same label encoding matrix C_Y across different tasks.

View consistency: For the i^{th} task, the decompositions of the instance-feature data X_{ij} $(1 \leq j \leq V)$ in different views share the same instance encoding matrix R_i.

Correlations among feature-instance-label: For the i^{th} task, the decompositions of instance-feature data X_{ij} and instance-label data Y_i share the same instance encoding matrix R_i.

Correlations among co-latent spaces: Since the instances, features, and labels may share the latent semantic concepts, we hope the learned co-latent spaces, M_{ij} and M_{iY}, are similar to each other.

The intuition of enhancing the correlations among co-latent spaces is as follows. Take webpage classification as an example. The words (1st view) on the webpage, the hyperlinks (2rd view) pointing to the webpage, and categories (labels) of webpage could be linked by the latent semantic topics (bridges) of the webpage. Therefore, we hope that the co-latent spaces $M_{ij}(1 \leq j \leq V)$ learned in the feature spaces from multiple views are as similar as possible to the co-latent space M_{iY} learned from label spaces, which acts as a bridge to link the labels with the features from multiple views in the latent spaces. Note that maximizing the correlations between co-latent spaces is equivalent to minimizing the distance between them.



(a) Multi-latent Space Model. (b) Hierarchical Multi-latent Space Model.

Fig. 1. An illustrative example about the proposed approach. In (a), without loss of generality, suppose there are two tasks and two views. The view consistency is modeled by sharing the instance encoding matrix R_1 (or R_2) across different views; The task relatedness is modeled by sharing the feature encoding matrices C_1 (or C_2) across different tasks; The label correlation is modeled by sharing the label encoding matrix C_Y across different tasks. In (b), the input data matrix $X_{ij}(1 \le i \le T, 1 \le j \le V)$ is decomposed into three matrices, $R_i^{(1)}$, $M_{ij}^{(1)}$, and $C_j^{(1)}$. Then, the co-latent space $M_{ij}^{(l-1)}$ is further decomposed to learn its own co-latent space $M_{ij}^{(l)}$ where $2 \le l \le L$. In such a way, the multi-latent space model can be used as a building block to stack up a multi-layer architecture in order to learn the hierarchical multi-latent space.

3.3 Hierarchical Multi-Latent Space Model

Motivated by the success of multi-layer models [19] in automatically extracting the hierarchical concepts from data, we use the multi-latent space model as a building block to stack up a multi-layer architecture. It aims to learn the hierarchical multi-latent space from complex heterogeneity.

The co-latent spaces M_{ij} and M_{iY} can be viewed as the compact representations for the original input data $\tilde{X}_{ij}$ and Y_i. Let L be the number of layers. For the co-latent space $M_{ij}^{(l-1)}$ (or $M_{iY}^{(l-1)}$) where $l(2 \le l \le L)$ represents the layer, we hope to further learn its own co-latent space $M_{ij}^{(l)}$ (or $M_{iY}^{(l)}$) in a higher level, i.e.,

$$M_{ij}^{(l-1)} \approx R_i^{(l)} M_{ij}^{(l)} C_j^{(l)T}$$

$$M_{iY}^{(l-1)} \approx R_i^{(l)} M_{iY}^{(l)} C_Y^{(l)T}$$

In such a way, we can gradually learn the factor matrices in each layer. Based on the learned co-latent spaces $M_{ij}^{(L)}$ and $M_{iY}^{(L)}$ in the highest layer L, we hope to recover the original input data, $\tilde{X}_{ij}$ and Y_i, in the first layer as accurately as possible. Thus, the objective for the multi-layer architecture is as follows:

$$\begin{aligned} \min_{\{R,M,C\}>0} & \sum_{i=1}^{T} \sum_{j=1}^{V} \mathcal{L}\left(\tilde{X}_{ij};\ \tilde{R}_i^{(1:L)} M_{ij}^{(L)} C_j^{(1:L)T}\right) \\ & + \alpha \sum_{i=1}^{T} \mathcal{L}\left(Y_i;\ R_i^{(1:L)} M_{iY}^{(L)} C_Y^{(1:L)T}\right) \\ & + \beta \sum_{i=1}^{T} \sum_{j=1}^{V} \mathcal{L}\left(M_{ij}^{(L)};\ M_{iY}^{(L)}\right) \end{aligned} \tag{2}$$

where $A^{(s:t)} = \prod_{l=s}^{t} A^{(l)}$ if $s \le t$, and $A^{(s:t)} = I$ otherwise for any matrix A. I is an identity matrix. For the simplicity of notation, we denote $\tilde{R}_i^{(1:L)} = \begin{bmatrix} R_i \\ R_i^u \end{bmatrix} R_i^{(2:L)}$.

Figure 1(b) shows an illustrative example for the proposed hierarchical multi-latent space model. Take the webpage classification or image annotation as the examples. In each layer, we do multi-way clustering on the instances, features and labels. Since the instances, features and labels may usually have hierarchical latent structures, they can be clustered into sub-categories, and further into high-level sub-categories, until the top categories. In such a way, we can gradually learn the more abstract semantic concepts in a higher layer.

3.4 Prediction

Note that the proposed hierarchical multi-latent space model works in a transductive fashion since the first term of Eq. 1 or Eq. 2 involves both training and test data in building the model.

After the model training, we can obtain the instance encoding matrices R_i^u for test data, R_i for training data, and $R_i^{(l)}(2 \le l \le L)$ shared by both training and test data. Then, we can use the factor matrices to predict the instances in the test data. The final prediction is the weighted sum of predictions resulting from each layer. We have $M_{iY}^{(l-1)} \approx R_i^{(l)} M_{iY}^{(l)} C_Y^{(l)T} (2 \le l \le L)$, and try to approximate Y_i^u by using $R_i^u M_{iY}^{(1)} C_Y^{(1)T}$. Therefore, the predicted instance-label matrix for the test data in $i^{th}(1 \le i \le T)$ task can be computed as follows:

$$F_i = \sum_{l=1}^{L} w_l F_i^{(l)} = \sum_{l=1}^{L} w_l R_i^u R_i^{(2:l)} M_{iY}^{(l)} C_Y^{(1:l)T} \tag{3}$$

where w_l controls the weight for l^{th} layer. A naïve way is to set the weights based on the reconstruction loss in each layer. In our experiments, we simply use the equal weight for each layer.

If the input instance-label matrix is a binary matrix, we can transform the predicted matrix F_i into a binary one by using 0.5 as the classification decision threshold.

3.5 Distance Metric

Various distance metric $\mathcal{L}(X;Y)$ can be used in our proposed model to measure the similarity between X and Y. In this paper, we focus on two divergence measures widely used in NMF models. One is the least squares loss function,

$$\|X-Y\|_F^2=\sum\nolimits_{i,j}\left(X_{ij}-Y_{ij}\right)^2$$

Another is the generalized Kullback-Leibler divergence,

$$D\left(X||\ Y\right)=\sum\nolimits_{i,j}\left(X_{ij}\log\frac{X_{ij}}{Y_{ij}}-X_{ij}+Y_{ij}\right)$$

It reduces to the Kullback-Leibler divergence when $\sum_{i,j}X_{ij}=\sum_{i,j}Y_{ij}=1$.

Note that both the least squares ($\beta=2$) and generalized KL divergence ($\beta=1$) are the special cases of β-divergence:

$$d_\beta\left(x|y\right)=\begin{cases}\frac{x}{y}-\log\frac{x}{y}-1 & \beta=0\\ x\log\frac{x}{y}-x+y & \beta=1\\ \frac{\left(x^\beta+(\beta-1)y^\beta-\beta xy^{\beta-1}\right)}{\beta(\beta-1)} & \beta\in R\backslash\{0,1\}\end{cases}$$

Next, we propose the optimization algorithm HiMLS based on least squares loss function in Section 4, and HiMLSD based on generalized Kullback-Leibler divergence in Section 5.

4 Optimization Algorithm for HiMLS

In this section, we introduce the two-phase optimization algorithm for HiMLS.

When using least squares loss function, the objective defined in Eq. 1 for **multi-latent space** can be instantiated as follows,

$$\begin{aligned}\min_{\{R,M,C\}>0}&\sum_{i=1}^{T}\sum_{j=1}^{V}\left\|\tilde{X}_{ij}-\tilde{R}_iM_{ij}C_j^T\right\|_F^2+\\ \alpha&\sum_{i=1}^{T}\left\|Y_i-R_iM_{iY}C_Y^T\right\|_F^2+\beta\sum_{i=1}^{T}\sum_{j=1}^{V}\left\|M_{ij}-M_{iY}\right\|_F^2\end{aligned}\tag{4}$$

The objective function defined in Eq. 2 for **hierarchical multi-latent space** can be instantiated as follows,

$$\begin{aligned}\min_{\{R,M,C\}>0}&\sum_{i=1}^{T}\sum_{j=1}^{V}\left\|\tilde{X}_{ij}-\tilde{R}_i^{(1:L)}M_{ij}^{(L)}C_j^{(1:L)T}\right\|_F^2\\ &+\alpha\sum_{i=1}^{T}\left\|Y_i-R_i^{(1:L)}M_{iY}^{(L)}C_Y^{(1:L)T}\right\|_F^2\\ &+\beta\sum_{i=1}^{T}\sum_{j=1}^{V}\left\|M_{ij}^{(L)}-M_{iY}^{(L)}\right\|_F^2\end{aligned}\tag{5}$$

Following the tactics successfully used in deep learning [19], we adopt a two-phase procedure to train the multi-layer model. We first pre-train the weights of each layer in a greedy layer-wise manner, then fine-tune the weights of all layers to reduce the total reconstruction loss and classification loss.

To derive the multiplicative update rules for pre-training (in Theorem 3) and fine-tuning (in Theorem 4) in HiMLS, we first derive Lemma 1. This lemma provides a generic method to derive the update rules for all of R, M, C in both pre-training and fine-tuning.

Lemma 1. *For any non-negative matrices M, X_i, R_i, C_i, P_j and K_j, the objective function,*

$$J\left(M\right)=\alpha\sum_i\left\|X_i-R_iMC_i^T\right\|_F^2+\beta\sum_j\left\|MP_j-K_j\right\|_F^2\tag{6}$$

is non-increasing under the update rule:

$$M=M\odot\sqrt{\frac{\alpha\sum_i R_i^TX_iC_i+\beta\sum_j K_jP_j^T}{\alpha\sum_i R_i^TR_iMC_i^TC_i+\beta\sum_j MP_jP_j^T}}\tag{7}$$

where α and β are the non-negative parameters.

Proof. We make use of auxiliary function approach [25] to derive the update rules for Eq. 6 and prove its convergence.

The objective function for M is rewritten into:

$$\begin{aligned}J\left(M\right)&=\alpha\sum_i\left\|X_i-R_iMC_i^T\right\|_F^2+\beta\sum_j\left\|MP_j-K_j\right\|_F^2\\ &=\alpha tr\sum_i\left[M^TR_i^TR_iMC_i^TC_i-2M^TR_i^TX_iC_i\right]\\ &\quad+\beta tr\sum_j\left[M^TMP_jP_j^T-2M^TK_jP_j^T\right]+const\end{aligned}$$

Let t be the index of iteration. Similar to [14], we can show that

$$\begin{aligned}G\left(M,M^{(t)}\right)=\alpha\sum_i\sum_{u,v}&\left\{\begin{array}{l}\frac{\left[R_i^TR_iM^{(t)}C_i^TC_i\right]_{uv}\cdot M_{uv}^2}{M_{uv}^{(t)}}-\\ 2\left[R_i^TX_iC_i\right]_{uv}M_{uv}^{(t)}\left(1+\ln\frac{M_{uv}}{M_{uv}^{(t)}}\right)\end{array}\right\}\\ +\beta\sum_j\sum_{u,v}&\left\{\begin{array}{l}\frac{\left[M^{(t)}P_jP_j^T\right]_{uv}\cdot M_{uv}^2}{M_{uv}^{(t)}}-\\ 2\left[K_jP_j^T\right]_{uv}M_{uv}^{(t)}\left(1+\ln\frac{M_{uv}}{M_{uv}^{(t)}}\right)\end{array}\right\}\end{aligned}$$

is an auxiliary function of $J\left(M\right)$ due to the facts:

$$G\left(M,M\right)=J\left(M\right)$$

and

$$G\left(M,M^{(t)}\right)\geq J\left(M\right).$$

The minimum is obtained by setting the derivative to zero:

$$\frac{\partial}{\partial M}G\left(M,M^{(t)}\right)=\mathbf{0}$$

Then, we get the update rule as follows:

$$M=M\odot\sqrt{\frac{\alpha\sum_i R_i^TX_iC_i+\beta\sum_j K_jP_j^T}{\alpha\sum_i R_i^TR_iMC_i^TC_i+\beta\sum_j MP_jP_j^T}}$$

Since $J\left(M^{(t)}\right)=G\left(M^{(t)},M^{(t)}\right)\geq\min_M G\left(M,M^{(t)}\right)=G\left(M^{(t+1)},M^{(t)}\right)\geq J\left(M^{(t+1)}\right)$, the objective function $J\left(M\right)$ is non-increasing under the above update rule. □

Lemma 2 shows that the iterative update method in Lemma 1 will converge to the stationary point.

Lemma 2. *The limiting solution of the update rule in Eq. 7 satisfies the KKT condition.*

Proof. For the function $J(M)$ in Eq. 6 with non-negative constraint, we introduce the Lagrangian function

$$L(M) = \alpha \sum_i \left\| X_i - R_i M C_i^T \right\|_F^2 + \beta \sum_j \|MP_j - K_j\|_F^2 - tr(\Lambda M^T)$$

where $\Lambda(\Lambda \geq \mathbf{0})$ is the Lagrangian multiplies matrix. The zero gradient condition gives

$$\frac{\partial L(M)}{\partial M} = \mathbf{0} \Rightarrow \Lambda = B - A$$

where

$$B = \alpha \sum\nolimits_i R_i^T R_i M C_i^T C_i + \beta \sum\nolimits_j M P_j P_j^T$$

$$A = \alpha \sum\nolimits_i R_i^T X_i C_i + \beta \sum\nolimits_j K_j P_j^T$$

According to the complementary slackness condition, we have

$$\Lambda \odot M = \mathbf{0} \Rightarrow (B - A) \odot M = \mathbf{0} \tag{8}$$

Next, we verify that the limiting solution of the update rule in Eq. 7 satisfies the above equation. When it converges, $M^{(\infty)} = M^{(t+1)} = M^{(t)} = M$ where t is the number of iteration, we have

$$(M \odot M) \odot B = (M \odot M) \odot A \Rightarrow (B - A) \odot (M \odot M) = \mathbf{0} \tag{9}$$

The equivalence between Eq. 8 and Eq. 9 completes the proof. □

Theorem 3 shows the multiplicative update rules for the multi-latent space model defined in Eq. 4, and demonstrates its convergence and correctness.

Theorem 3 (Convergence of Pre-training)**.** *The objective function in Eq. 4 is non-increasing under the update rules:*

$$R_i = R_i \odot \sqrt{\frac{\sum_{j=1}^{V} X_{ij} C_j M_{ij}^T + \alpha Y_i C_Y M_{iY}^T}{\sum_{j=1}^{V} R_i M_{ij} C_j^T C_j M_{ij}^T + \alpha R_i M_{iY} C_Y^T C_Y M_{iY}^T}} \tag{10}$$

$$R_i^u = R_i^u \odot \sqrt{\frac{\sum_{j=1}^{V} X_{ij}^u C_j M_{ij}^T}{\sum_{j=1}^{V} R_i^u M_{ij} C_j^T C_j M_{ij}^T}} \tag{11}$$

$$C_j = C_j \odot \sqrt{\frac{\sum_{i=1}^{T} \tilde{X}_{ij}^T \tilde{R}_i M_{ij}}{\sum_{i=1}^{T} C_j M_{ij}^T \tilde{R}_i^T \tilde{R}_i M_{ij}}} \tag{12}$$

$$C_Y = C_Y \odot \sqrt{\frac{\sum_{i=1}^{T} Y_i^T R_i M_{iY}}{\sum_{i=1}^{T} C_Y M_{iY}^T R_i^T R_i M_{iY}}} \tag{13}$$

$$M_{ij} = M_{ij} \odot \sqrt{\frac{\tilde{R}_i^T \tilde{X}_{ij} C_j + \beta M_{iY}}{\tilde{R}_i^T \tilde{R}_i M_{ij} C_j^T C_j + \beta M_{ij}}} \tag{14}$$

$$M_{iY} = M_{iY} \odot \sqrt{\frac{\alpha R_i^T Y_i C_Y + \beta \sum_{j=1}^{V} M_{ij}}{\alpha R_i^T R_i M_{iY} C_Y^T C_Y + \beta V M_{iY}}} \tag{15}$$

Also, the limiting solutions of the update rules satisfy the KKT condition.

Proof. The convergence of the update follows from Lemma 1. According to Lemma 2, we can prove that the limiting solutions satisfy the KKT condition. □

For simplicity, denote $\Omega_{ij} = R_i^{(1:L)} M_{ij}^{(L)} C_j^{(1:L)T}$, $\tilde{\Omega}_{ij} = \tilde{R}_i^{(1:L)} M_{ij}^{(L)} C_j^{(1:L)T}$, $\Omega_{iY} = R_i^{(1:L)} M_{iY}^{(L)} C_Y^{(1:L)T}$, and $\Phi(A) = R_i^{(1:l-1)T} A R_i^{(l+1:L)T}$ for any matrix A.

Theorem 4 shows the multiplicative update rules for the hierarchical multi-latent space model defined in Eq. 5, and demonstrates its convergence and correctness.

Theorem 4 (Convergence of Fine-tuning)**.** *The objective function in Eq. 5 is non-increasing under the update rules:*

$$R_i^{(l)} = R_i^{(l)} \odot \sqrt{\frac{\sum_{j=1}^{V} \Phi\left(X_{ij} C_j^{(1:L)} M_{ij}^{(L)T}\right) + \alpha \Phi\left(Y_i C_Y^{(1:L)} M_{iY}^{(L)T}\right)}{\sum_{j=1}^{V} \Phi\left(\Omega_{ij} C_j^{(1:L)} M_{ij}^{(L)T}\right) + \alpha \Phi\left(\Omega_{iY} C_Y^{(1:L)} M_{iY}^{(L)T}\right)}} \tag{16}$$

$$R_i^u = R_i^u \odot \sqrt{\frac{\sum_{j=1}^{V} X_{ij}^u C_j^{(1:L)} M_{ij}^{(L)T} R_i^{(2:L)T}}{\sum_{j=1}^{V} R_i^u R_i^{(2:L)} M_{ij}^{(L)} C_j^{(1:L)T} C_j^{(1:L)} M_{ij}^{(L)T} R_i^{(2:L)T}}} \tag{17}$$

$$C_j^{(l)} = C_j^{(l)} \odot \sqrt{\frac{\sum_{i=1}^{T} C_j^{(1:l-1)T} \tilde{X}_{ij}^T \tilde{R}_i^{(1:L)} M_{ij}^{(L)} C_j^{(l+1:L)T}}{\sum_{i=1}^{T} C_j^{(1:l-1)T} \tilde{\Omega}_{ij}^T \tilde{R}_i^{(1:L)} M_{ij}^{(L)} C_j^{(l+1:L)T}}} \tag{18}$$

$$C_Y^{(l)} = C_Y^{(l)} \odot \sqrt{\frac{\sum_{i=1}^{T} C_Y^{(1:l-1)T} Y_i^T R_i^{(1:L)} M_{iY}^{(L)} C_Y^{(l+1:L)T}}{\sum_{i=1}^{T} C_Y^{(1:l-1)T} \Omega_{iY}^T R_i^{(1:L)} M_{iY}^{(L)} C_Y^{(l+1:L)T}}} \tag{19}$$

$$M_{ij}^{(L)} = M_{ij}^{(L)} \odot \sqrt{\frac{\tilde{R}_i^{(1:L)T} \tilde{X}_{ij} C_j^{(1:L)} + \beta M_{iY}^{(L)}}{\tilde{R}_i^{(1:L)T} \tilde{\Omega}_{ij} C_j^{(1:L)} + \beta M_{ij}^{(L)}}} \tag{20}$$

$$M_{iY}^{(L)} = M_{iY}^{(L)} \odot \sqrt{\frac{\alpha R_i^{(1:L)T} Y_i C_Y^{(1:L)} + \beta \sum_{j=1}^{V} M_{ij}^{(L)}}{\alpha R_i^{(1:L)T} \Omega_{iY} C_Y^{(1:L)} + \beta V M_{iY}^{(L)}}} \tag{21}$$

where $1 \leq l \leq L$. *Also, the limiting solutions of the update rules satisfy the KKT condition.*

Proof. According to Lemma 1, we can prove the convergence of the updating. According to Lemma 2, we can prove that the limiting solutions satisfy the KKT condition. □

After obtaining $R_i^{(l)}$ and $C_j^{(l)}$, we can update $M_{ij}^{(l)}$ and $M_{iY}^{(l)}$ $(1 \leq l < L)$ by using the update rule got in the pre-training phase.

Based on Theorem 3 and Theorem 4, we summarize the optimization algorithm for HiMLS in Algorithm 1. There are two training phases including pre-training and fine-tuning

in Algorithm 1. As shown in Steps 1-9, the pre-training phase goes forward from the first layer to the highest layer, and each layer is trained in a greedy layer-wise manner. In contrast, the fine-tuning phase shown in Steps 10-17 moves in an opposite direction, and the weights of all the layers will be updated. The convergence of the HiMLS algorithm is guaranteed by Theorem 3 and Theorem 4.

Algorithm 1 HiMLS Algorithm

Input: Instance-feature matrices $\tilde{X}_{ij}\,(1 \le i \le T, 1 \le j \le V)$, instance-label matrices for train data $Y_i\,(1 \le i \le T)$, α, β, number of layers L.
Output: Predicted instance-label matrices $F_i\,(1 \le i \le T)$ for test data.

1: **for** $l = 1 : L$ **do**
2: Initialize $\tilde{R}_i^{(l)}\,(1 \le i \le T)$, $C_Y^{(l)}$ and $C_j^{(l)}\,(1 \le j \le V)$ by clustering instances, labels, and features using probabilistic latent semantic analysis, respectively;
3: Initialize $M_{ij}^{(l)} = R_i^{(l)\dagger} M_{ij}^{(l-1)} C_j^{(l)\dagger}$, $M_{iY}^{(l)} = R_i^{(l)\dagger} M_{iY}^{(l-1)} C_Y^{(l)\dagger}$ where $R^\dagger = \left(R^T R\right)^{-1} R^T$ and $C^\dagger = C\left(C^T C\right)^{-1}$. Note that $M_{ij}^{(0)} = \tilde{X}_{ij}$, and $M_{iY}^{(0)} = Y_i$;
4: **repeat**
5: Update $R_i^{(l)}\,(1 \le i \le T)$ and R_i^u by Eq. 10 and Eq. 11;
6: Update $C_j^{(l)}\,(1 \le j \le V)$ and $C_Y^{(l)}$ by Eq. 12 and Eq. 13;
7: Update $M_{ij}^{(l)}$ and $M_{iY}^{(l)}$ where $1 \le i \le T, 1 \le j \le V$ by Eq. 14 and Eq. 15;
8: **until** converged
9: **end for**;
10: **repeat**
11: Update $M_{ij}^{(L)}$ and $M_{iY}^{(L)}$ where $1 \le i \le T, 1 \le j \le V$ by Eq. 20 and Eq. 21;
12: **for** $l = L : 1$ **do**
13: Update $R_i^{(l)}\,(1 \le i \le T)$ and R_i^u by Eq. 16 and Eq. 17;
14: Update $C_j^{(l)}\,(1 \le j \le V)$ and $C_Y^{(l)}$ by Eq. 18 and Eq. 19;
15: Update $M_{ij}^{(l)}$ and $M_{iY}^{(l)}$ where $1 \le i \le T, 1 \le j \le V, l \ne L$ by Eq. 14 and Eq. 15;
16: **end for**;
17: **until** converged
18: **return** Predictions for the test data using Eq. 3.

Time complexity: Similar to other matrix factorization methods based on multiplicative update rules [14], [25], a nice property of the proposed HiMLS algorithm is that most of the computations are matrix multiplications and can be computed efficiently. Lemma 5 shows the complexity of the algorithm. The proof is omitted for brevity.

Lemma 5 (Complexity)**.** *The time complexity for the multiplicative update rules in Theorem 3 are as follows:*

$$\mathcal{O}(R_i) = \mathcal{O}(R_i^u) = \mathcal{O}\left(\sum\nolimits_{j=1}^{V} n_i N\left(pq + q^2 + d_j q + mq\right)\right)$$

$$\mathcal{O}(C_j) = \mathcal{O}\left(\sum\nolimits_{i=1}^{T} d_j N\left(n_i p + pq + p^2\right)\right)$$

$$\mathcal{O}(C_Y) = \mathcal{O}\left(\sum\nolimits_{i=1}^{T} mN\left(n_i p + pq + p^2\right)\right)$$

$$\mathcal{O}(M_{ij}) = \mathcal{O}\left(pN\left(n_i d_j + q d_j + pq + q^2\right)\right)$$

$$\mathcal{O}(M_{iY}) = \mathcal{O}\left(pN\left(n_i m + qm + pq + q^2\right)\right)$$

where $1 \le i \le T, 1 \le j \le V$ and N is the number of iteration until convergence.

Note that the dimensions of the latent spaces are usually far smaller than the ones in the original spaces, i.e., $p \ll n_i$ and $q \ll d_j$. Lemma 5 shows that the multiplicative update rules for pre-training are scalable to the problem sizes. Likewise, we can obtain the time complexity of the update rules for fine-tuning, which are omitted for brevity.

5 Optimization Algorithm for HiMLSD

In this section, we introduce the optimization algorithm for HiMLSD, which is the counterpart of HiMLS.

HiMLSD adopts the generalized Kullback-Leibler divergence (see subsection 3.5) as loss metric. Therefore, the objective function defined in Eq. 1 for **multi-latent space** is instantiated as,

$$\begin{aligned}\min_{\{R,M,C\}>0} &\sum_{i=1}^{T}\sum_{j=1}^{V} D\left(\tilde{X}_{ij}||\tilde{R}_i M_{ij} C_j^T\right) \\ &+ \alpha\sum_{i=1}^{T} D\left(Y_i||R_i M_{iY} C_Y^T\right) + \beta\sum_{i=1}^{T}\sum_{j=1}^{V} D\left(M_{ij}||M_{iY}\right)\end{aligned} \tag{22}$$

The objective function defined in Eq. 2 for **hierarchical multi-latent space** can be instantiated as,

$$\begin{aligned}\min_{\{R,M,C\}>0} &\sum_{i=1}^{T}\sum_{j=1}^{V} D\left(\tilde{X}_{ij}||\tilde{R}_i^{(1:L)} M_{ij}^{(L)} C_j^{(1:L)T}\right) \\ &+ \alpha\sum_{i=1}^{T} D\left(Y_i||R_i^{(1:L)} M_{iY}^{(L)} C_Y^{(1:L)T}\right) \\ &+ \beta\sum_{i=1}^{T}\sum_{j=1}^{V} D\left(M_{ij}^{(L)}||M_{iY}^{(L)}\right)\end{aligned} \tag{23}$$

Next, we derive Lemma 6, which is a generic method to derive the multiplicative update rules for R, M, C in both pre-training and fine-tuning of HiMLDS.

Lemma 6. *For any non-negative matrices H, X, R, C and P, the function*

$$F(H) = \alpha D\left(X||RHC^T\right) + \beta D\left(H||P\right) \tag{24}$$

is non-increasing under the update:

$$H \leftarrow \frac{H \odot \left(\alpha R^T \frac{X}{RHC^T} C\right) + \beta P}{\beta E + \alpha R^T E C} \tag{25}$$

where α and β are non-negative parameters. E is a unit matrix whose dimensions are set wherever appropriate.

Proof. The function $F(H)$ can be rewritten as,

$$\begin{aligned}F(h) &= \alpha D\left(X||RHC^T\right) + \beta D\left(H||P\right) \\ &= \alpha D\left(vec\left(X\right)||vec\left(RHC^T\right)\right) + \beta D\left(vec(H)||vec(P)\right) \\ &= \alpha D\left(vec\left(X\right)||\left(C \otimes R\right)vec\left(H\right)\right) + \beta D\left(vec\left(H\right)||vec\left(P\right)\right) \\ &= \alpha D\left(x||Wh\right) + \beta D\left(h||p\right) \\ &= \alpha\sum_i\left(x_i \log\frac{x_i}{\sum_j W_{ij}h_j} - x_i + \sum\nolimits_j W_{ij}h_j\right) \\ &\quad + \beta\sum_j\left(h_j \log\frac{h_j}{p_j} - h_j + p_j\right)\end{aligned}$$

where $W = C \otimes R$. The key issue is to design an auxiliary function for $F(h)$. Denote

$$G\left(h, h^t\right) = \alpha \sum_i \left(x_i \log x_i - x_i + \sum\nolimits_j W_{ij} h_j\right) - \alpha \sum_{i,j} x_i \frac{W_{ij} h_j^t}{\sum_k W_{ik} h_k^t} \left(\log W_{ij} h_j - \log \frac{W_{ij} h_j^t}{\sum_k W_{ik} h_k^t}\right) + \beta \sum_j \left(h_j \log \frac{h_j}{p_j} - h_j + p_j\right)$$

To show that $G\left(h, h^t\right)$ is an auxiliary function of $F(h)$, we need to prove: (1) $G\left(h, h\right) = F\left(h\right)$; (2) $G\left(h, h^t\right) \geq F\left(h\right)$. The first equation is straightforward. To prove the latter inequality, we use the convexity of log function:

$$-\log \sum_j W_{ij} h_j \leq -\sum_j c_j \log \frac{W_{ij} h_j}{c_j}$$
$$s.t. \quad c_j \geq 0, \quad \sum_j c_j = 1$$

Setting $c_j = \frac{W_{ij} h_j^t}{\sum_k W_{ik} h_k^t}$, we obtain,

$$-\log \sum_j W_{ij} h_j \leq -\sum_j \frac{W_{ij} h_j^t}{\sum_k W_{ik} h_k^t} \log \frac{W_{ij} h_j \sum_k W_{ik} h_k^t}{W_{ij} h_j^t}$$

From this inequality it follows that $G\left(h, h^t\right) \geq F\left(h\right)$.

The minimum of $G\left(h, h^t\right)$ with respect to h is determined by setting the gradient to zero:

$$\frac{dG\left(h, h^t\right)}{dh_j} = -\alpha \sum_i \left(\frac{x_i W_{ij} h_j^t}{h_j \sum_k W_{ik} h_k^t} - W_{ij}\right) + \beta \log \frac{h_j}{p_j} = 0$$

Since $\log x \approx 1 - 1/x$ with $x \to 1$ [6], the above equation can be approximated as:

$$-\alpha \sum_i \left(\frac{x_i W_{ij} h_j^t}{h_j \sum_k W_{ik} h_k^t} - W_{ij}\right) + \beta \left(1 - \frac{p_j}{h_j}\right) = 0 \quad (26)$$

According to Eq. 26, we have,

$$h_j = \frac{h_j^t \sum_i \frac{\alpha x_i W_{ij}}{\sum_k W_{ik} h_k^t} + \beta p_j}{\beta + \alpha \sum_i W_{ij}}$$
$$\Rightarrow h = \frac{h^t \odot \left(\alpha W^T \frac{x}{Wh}\right) + \beta p}{\alpha W^T \cdot \mathbf{1} + \beta \cdot \mathbf{1}} = \frac{h^t \odot \left(\alpha \left(C^T \otimes R^T\right) vec\left(\frac{X}{RHC^T}\right)\right) + \beta p}{\alpha \cdot vec\left(R^T E C\right) + \beta \cdot \mathbf{1}} = \frac{h^t \odot vec\left(\alpha R^T \frac{X}{RHC^T} C\right) + \beta vec\left(P\right)}{\alpha \cdot vec\left(R^T E C\right) + \beta vec\left(E\right)}$$
$$\Rightarrow H \leftarrow \frac{H \odot \left(\alpha R^T \frac{X}{RHC^T} C\right) + \beta P}{\alpha R^T E C + \beta E}$$

□

Similar to the proof in Lemma 6, we can derive the following lemma.

Lemma 7. *The function*

$$J\left(H\right) = \beta D\left(H || P\right) + \sum_k \alpha_k D\left(X_k || R_k H C_k^T\right) \quad (27)$$

is non-increasing under the update rule:

$$H \leftarrow \frac{H \odot \left(\sum_k \alpha_k R_k^T \frac{X_k}{R_k H C_k^T} C_k\right) + \beta P}{\beta E + \sum_k \alpha_k R_k^T E_k C_k} \quad (28)$$

where β and α_k are non-negative parameters. E (or E_k) is a unit matrix whose dimensions are set wherever appropriate.

Lemma 8 shows that the iterative update method in Lemma 6 will converge to the stationary point.

Lemma 8. *The limiting solution of the update rule in Eq. 25 satisfies the KKT condition.*

Proof. For the function $F(H)$ in Eq. 24 with non-negative constraint, we introduce the Lagrangian function

$$L\left(h\right) = \alpha \sum_i \left(x_i \log \frac{x_i}{\sum_j W_{ij} h_j} - x_i + \sum\nolimits_j W_{ij} h_j\right) + \beta \sum_j \left(h_j \log \frac{h_j}{p_j} - h_j + p_j\right) - tr(\Lambda h^T)$$

where $\Lambda (\Lambda \geq \mathbf{0})$ is the Lagrangian multiplies vector. The zero gradient condition gives

$$\frac{\partial L(h)}{\partial h_j} = 0 \Rightarrow \Lambda_j = \alpha \sum_i W_{ij} \left(1 - \frac{x_i}{\sum_k W_{ik} h_k}\right) + \beta \log \frac{h_j}{p_j}$$

According to the complementary slackness condition $\Lambda_j \odot h_j = 0$, we have

$$\left[\alpha \sum_i W_{ij} \left(1 - \frac{x_i}{\sum_k W_{ik} h_k}\right) + \beta \log \frac{h_j}{p_j}\right] \cdot h_j = 0 \quad (29)$$

Likewise, when $x \to 1$, $\log x \approx 1 - 1/x$, the above equation can be approximated as:

$$\left[\alpha \sum_i W_{ij} \left(1 - \frac{x_i}{\sum_k W_{ik} h_k}\right) + \beta \left(1 - \frac{p_j}{h_j}\right)\right] \cdot h_j = 0 \quad (30)$$

Next, we verify that the limiting solution of the update rule in Eq. 25 satisfies the above equation. When it converges, $h_j^{(\infty)} = h_j^{(t+1)} = h_j^{(t)} = h_j$ where t is the index of iteration, we have

$$h_j \left(\beta + \alpha \sum_i W_{ij}\right) = h_j \sum_i \frac{\alpha x_i W_{ij}}{\sum_k W_{ik} h_k} + \beta p_j \quad (31)$$

The equivalence between Eq. 30 and Eq. 31 completes the proof. □

Next we derive the multiplicative update rules for pre-training and fine-tuning in HiMLDS. Theorem 9 shows the multiplicative update rules for Eq. 22, and demonstrates its convergence and correctness.

Theorem 9 (Convergence of Pre-training)**.** *The objective function in Eq. 22 is non-increasing under the update rules:*

$$M_{ij} \leftarrow \frac{M_{ij} \odot \left(\tilde{R}_i^T \frac{\tilde{X}_{ij}}{\tilde{R}_i M_{ij} C_j^T} C_j\right) + \beta M_{iY}}{\tilde{R}_i^T E_{ij} C_j + \beta E} \quad (32)$$

$$M_{iY} \leftarrow \frac{M_{iY} \odot \left(\alpha R_i^T \frac{Y_i}{R_i M_{iY} C_Y^T} C_Y + \beta \sum_{j=1}^{V} \frac{M_{ij}}{M_{iY}} \right)}{\alpha R_i^T E_i C_Y + \beta V \cdot E} \tag{33}$$

$$R_i \leftarrow \frac{R_i \odot \left(\alpha \frac{Y_i}{R_i M_{iY} C_Y^T} C_Y M_{iY}^T + \sum_{j=1}^{V} \frac{X_{ij}}{R_i M_{ij} C_j^T} C_j M_{ij}^T \right)}{\alpha E_i C_Y M_{iY}^T + \sum_{j=1}^{V} E_{ij} C_j M_{ij}^T} \tag{34}$$

$$R_i^u \leftarrow \frac{R_i^u \odot \sum_{j=1}^{V} \frac{X_{ij}^u}{R_i^u M_{ij} C_j^T} C_j M_{ij}^T}{\sum_{j=1}^{V} E_{ij} C_j M_{ij}^T} \tag{35}$$

$$C_j^T \leftarrow \frac{C_j^T \odot \sum_{i=1}^{T} M_{ij}^T \tilde{R}_i^T \frac{\tilde{X}_{ij}}{\tilde{R}_i M_{ij} C_j^T}}{\sum_{i=1}^{T} M_{ij}^T \tilde{R}_i^T E_{ij}} \tag{36}$$

$$C_Y^T \leftarrow \frac{C_Y^T \odot \sum_{i=1}^{T} M_{iY}^T R_i^T \frac{Y_i}{R_i M_{iY} C_Y^T}}{\sum_{i=1}^{T} M_{iY}^T R_i^T E_{ij}} \tag{37}$$

Proof. According to Lemma 7, we can prove the convergence of the updating. According to Lemma 8, we can prove that the limiting solutions satisfy the KKT condition. □

Define $\Omega_{ij}(l) = R_i^{(l+1:L)} M_{ij}^{(L)} C_j^{(1:L)T}$, $\Omega_{iY}(l) = R_i^{(l+1:L)} M_{iY}^{(L)} C_Y^{(1:L)T}$, $\Theta_{ij}(l) = \tilde{R}_i^{(1:L)} M_{ij}^{(L)} C_j^{(l+1:L)T}$, and $\Theta_{iY}(l) = R_i^{(1:L)} M_{iY}^{(L)} C_Y^{(l+1:L)T}$.

Theorem 10 shows the multiplicative update rules for Eq. 23, and demonstrates its convergence and correctness.

Theorem 10 (Convergence of Fine-tuning)**.** *The objective function in Eq. 23 is non-increasing under the update rules:*

$$M_{ij}^{(L)} \leftarrow \frac{M_{ij}^{(L)} \odot \left(\tilde{R}_i^{(1:L)T} \frac{\tilde{X}_{ij}}{\tilde{R}_i^{(1:L)} M_{ij}^{(L)} C_j^{(1:L)T}} C_j^{(1:L)} \right) + \beta M_{iY}^{(L)}}{\tilde{R}_i^{(1:L)T} E_{ij} C_j^{(1:L)} + \beta E} \tag{38}$$

$$M_{iY}^{(L)} \leftarrow \frac{M_{iY}^{(L)} \odot \left(\alpha R_i^{(1:L)T} \frac{Y_i}{R_i^{(1:L)} M_{iY}^{(L)} C_Y^{(1:L)T}} C_Y^{(1:L)} + \beta \sum_{j=1}^{V} \frac{M_{ij}^{(L)}}{M_{iY}^{(L)}} \right)}{\alpha R_i^{(1:L)T} E_i C_Y^{(1:L)} + \beta V \cdot E} \tag{39}$$

$$R_i^{(l)} \leftarrow \frac{R_i^{(l)} \odot \left(\alpha R_i^{(1:l-1)T} \frac{Y_i}{\Omega_{iY}(0)} \Omega_{iY}^T(l) + \sum_{j=1}^{V} R_i^{(1:l-1)T} \frac{X_{ij}}{\Omega_{ij}(0)} \Omega_{ij}^T(l) \right)}{\alpha R_i^{(1:l-1)T} E_i \Omega_{iY}^T(l) + \sum_{j=1}^{V} R_i^{(1:l-1)T} E_{ij} \Omega_{ij}^T(l)} \tag{40}$$

$$R_i^u \leftarrow \frac{R_i^u \odot \left(\sum_{j=1}^{V} \frac{X_{ij}^u}{R_i^u \Omega_{ij}(1)} \Omega_{ij}^T(1) \right)}{\sum_{j=1}^{V} E_{ij} \Omega_{ij}^T(1)} \tag{41}$$

$$C_j^{(l)T} \leftarrow \frac{C_j^{(l)T} \odot \sum_{i=1}^{T} \Theta_{ij}^T(l) \frac{\tilde{X}_{ij}}{\tilde{R}_i^{(1:L)} M_{ij}^{(L)} C_j^{(1:L)T}} C_j^{(1:l-1)}}{\sum_{i=1}^{T} \Theta_{ij}^T(l) E_{ij} C_j^{(1:l-1)}} \tag{42}$$

$$C_Y^{(l)T} \leftarrow \frac{C_Y^{(l)T} \odot \sum_{i=1}^{T} \Theta_{iY}^T(l) \frac{Y_i}{R_i^{(1:L)} M_{iY}^{(L)} C_Y^{(1:L)T}} C_Y^{(1:l-1)}}{\sum_{i=1}^{T} \Theta_{iY}^T(l) E_i C_Y^{(1:l-1)}} \tag{43}$$

where $1 \le l \le L$. Also, the limiting solutions of the update rules satisfy the KKT condition.

Proof. According to Lemma 7, we can prove the convergence of the updating. According to Lemma 8, we can prove that the limiting solutions satisfy the KKT condition. □

Likewise, we can obtain the algorithm for HiMLSD and its time complexity, which are omitted for brevity.

6 The Special Cases of HiMLS

The proposed model is a generalized framework for learning complex heterogeneity. It is widely applicable to multiple types of heterogeneous learning problems.

A special case of HiMLS is to learn the common co-latent space M shared among all the tasks, view, and labels, i.e., $M_{ij} = M_{iY} = M (1 \le i \le T, 1 \le j \le V)$. And by using the training data only, Eq. 4 can be specialized as:

$$\min_{R,M,C} \sum_{i=1}^{T} \sum_{j=1}^{V} \left\| X_{ij} - R_i M C_j^T \right\|_F^2 + \alpha \sum_{i=1}^{T} \left\| Y_i - R_i M C_Y^T \right\|_F^2 \tag{44}$$

It is worth noting that Eq. 44 is not a trivial special case. Theorem 11 shows that some popular methods for learning from single heterogeneity can be viewed as the special cases of our proposed model, such as the multi-view learning method MSL [36] and the multi-label learning method LS-CCA [32]. Both MSL and LS-CCA are closely related to canonical correlation analysis (CCA), while MSL is an unsupervised learning method aiming to learn the subspace from multiple views, and LS-CCA is a supervised learning method for the multi-label problem when one of the views used in CCA is derived from the labels.

Theorem 11. *The multi-view learning method MSL [36] and the multi-label learning method LS-CCA [32] can be viewed as the special cases of HiMLS.*

Proof. Consider two special cases of HiMLS for learning from a single heterogeneity as follows:

1) Unsupervised multi-view learning: By letting $T = 1$, $V = 2$, and $\alpha = 0$, Eq. 44 can be rewritten into:

$$\min_{R,M,C} \left\| X_1 - R M C_1^T \right\|_F^2 + \left\| X_2 - R M C_2^T \right\|_F^2 \tag{45}$$

where $X_j (j = 1, 2)$ is the instance-feature matrix for the j^{th} view.

2) Supervised multi-label learning: By letting $T = 1, V = 1$, and $\alpha = 1$, Eq. 44 can be rewritten into:

$$\min_{R,M,C} \left\| X - R M C_1^T \right\|_F^2 + \left\| Y - R M C_2^T \right\|_F^2 \tag{46}$$

where X and Y are the instance-feature matrix and instance-label matrix, respectively.

Both Eq. 45 and Eq. 46 have the same form as follows:

$$\min_{H,C} \left\| [X, Y] - H C^T \right\|_F^2 \tag{47}$$

where $H = RM$ and $C^T = [C_1^T, C_2^T]$. Consider the normalized data matrix defined as $Z = \left[(XX^T)^{-\frac{1}{2}} X, (YY^T)^{-\frac{1}{2}} Y \right]$. When imposing the orthogonal constraint $C^T C = I$, Eq. 47 can be rewritten into:

$$\min_{H, C^T C = I} \left\| Z - H C^T \right\|_F^2 \tag{48}$$

Let $f(H,C)$ denote the objective function for Eq. 48, which can be transformed into:

$$f(H,C) = tr\left[Z^TZ - 2CH^TZ + H^TH\right]$$

When fixing C, we have:

$$\nabla_H f(H,C) = -2ZC + 2H = 0 \Rightarrow H = ZC$$

By substituting $H = ZC$ into Eq. 48, we have

$$\begin{aligned}&\min_{C^TC=I} \left\|Z - HC^T\right\|_F^2 \\ &= \min_{C^TC=I} tr\left[Z^TZ - 2CC^TZ^TZ + C^TZ^TXC\right] \\ &= tr\left[Z^TZ\right] - \max_{C^TC=I} tr\left[C^TZ^TZC\right]\end{aligned} \quad (49)$$

The optimal solution for C is given by the top k eigenvectors of Z^TZ. According to [36], Eq. 49 has the same optimal solution with CCA which aims to optimize:

$$\max_{U,V} tr\left(U^TXY^TV\right) \quad s.t. \quad U^TXX^TU = V^TYY^TV = I$$

Therefore, the first special case of HiMLS is equivalent to applying CCA to the instance-feature matrices from multiple views [36]. The second special case of HiMLS is equivalent to applying CCA to both the instance-feature matrix and instance-label matrix [32]. □

7 Experiments

In this section, we demonstrate the effectiveness of the proposed algorithms on various data sets in comparison with different heterogeneous learning methods.

7.1 Data Sets and Setup

Four real data sets from different domains are used for evaluation, including text, image, and manufacturing data.

The first data set is the Reuters Corpus Volume I (RCV1V2) [1] data set [26], which is a collection of over 800,000 newswire stories. There are three category sets of data: Topics (i.e. major subject of a story), Industry Codes (i.e. type of business discussed), and Regions (i. e. geographic locations). Each of these category sets has a hierarchical structures. It is usually common to use several subsets of this data, each containing 6000 data instances on average and with a total number of 101 class labels.

EUR-Lex [27] is a text data set containing European Union official laws in practice, different kinds of treaties and agreements, parliamentary journals. This data set contains nearly 20,000 text documents classified according to three different schemas: i) subject matter (e.g. agriculture), ii) official classification hierarchy called the directory codes (e.g. a document belonging to a class also belongs to all its parent classes), and iii) EUROVOC, a multilingual thesaurus maintained by the Office for Official Publications of the European Communities. Each of these category sets forms a hierarchical structures.

NUS-WIDE [2] [9] is the a real-world web image data set comprising over 269,000 images with over 5,000 user-provided tags, and ground-truth of 81 concepts with a hierarchical structures. There are several types of low-level visual features such as 64-D color histogram in LAB color space, 144-D color correlogram in HSV color space, 73-D edge distribution histogram, and 500-D bag of visual words. We use the light version of NUS-WIDE.

In these data sets, the label refer to the multiple categories each instance belonging to. For the NUS-WIDE data, the view refers to different types of low-level visual feature. For either RCV1V2 or EUR-Lex data sets, similar to [42], the data are described from two views: one corresponds to the TF-IDF features; another corresponds to the latent topics obtained by applying probabilistic latent semantic analysis[3] on the term counts. The task refers to classify the instances belonging to different sub-categories, which follow different but related distributions [18].

The last data set AL-SMELT is related to manufacturing process. AL-SMELT is collected from Aluminum smelting process. This data set corresponds to an electrolytic process with 174 process variables that forms 4 views based on the process control practice: power and resistance, noise control, feed control, and chemicals. It is concurrently running in 245 smelters, which can be classified into 5 groups (tasks) based on their design and generation. The 174 process variables are collected automatically at daily level via sensors. Two other important control variables, temperature and Alumina Fluoride, are collected every other day manually. Here, the goal is to predict the change direction (increase or decrease) of these 2 variables (labels) when they are not collected. The prediction fills in the information gap and enables feedback control in a finer granularity.

Table 1 shows the properties of different data sets. Label cardinality is the average number of labels per instance. Accordingly, label density normalizes label cardinality by the the number of labels. Label diversity is the number of distinct label combinations observed in the data set [46].

7.2 Evaluation Metrics

In order to comprehensively investigate the performance of the proposed method, we use F_1-score, accuracy and Hamming loss on the test data as the evaluation metrics.

F_1-score [46] is the harmonic mean of precision and recall where precision is the proportion of predicted correct labels to the total number of actual labels, recall is the proportion of predicted correct labels to the total number of predicted labels, averaged over all instances. Note that the larger value of F_1-score is indicating the better performance.

Accuracy [46] for each instance is defined as the proportion of the predicted correct labels to the total number of labels for that instance. Overall accuracy is the average across all instances. Note that the larger value of accuracy is indicating the better performance.

Hamming Loss [46] reports how many times on average, the relevance of an instance to a class label is incorrectly predicted. Therefore, hamming loss takes into account the prediction error (an incorrect label is predicted) and the missing error (a relevant label not predicted), normalized over total number of classes and total number of instances. Note that the smaller the value of Hamming loss, the better the performance of the learning algorithm.

1. http://mulan.sourceforge.net/datasets-mlc.html
2. http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm
3. http://lear.inrialpes.fr/people/verbeek/code

TABLE 1
Statistics of Different Data Sets.

Data set	Instances	Features	Labels	Cardinality	Density	Diversity
RCV1V2_1	6000	47236	101	2.880	0.029	1028
RCV1V2_2	6000	47236	101	2.634	0.026	954
RCV1V2_3	6000	47229	101	2.614	0.026	939
RCV1V2_4	6000	47236	101	2.484	0.025	816
EUR-Lex	19348	5000	412	1.292	0.003	1615
NUS-WIDE	55615	708	81	1.869	0.023	18430
AL-SMELT	6468	174	2	0.986	0.493	4

TABLE 2
Comparison among HiMLS Variants on RCV1V2_1 Data Set.

ALGORITHM	F_1-SCORE	ACCURACY	HAMMING LOSS
HIMLS	**.906±.006**	**.885±.006**	**.064±.003**
MLS	.868±.023	.829±.028	.102±.018
MLS-T	.864±.008	.822±.009	.106±.006
MLS-V	.857±.009	.813±.010	.107±.005
MLS-S	.847±.018	.805±.020	.116±.011

TABLE 3
Comparison among HiMLS Variants on RCV1V2_2 Data Set.

ALGORITHM	F_1-SCORE	ACCURACY	HAMMING LOSS
HIMLS	**.903±.006**	**.872±.006**	**.073±.003**
MLS	.880±.006	.846±.008	.089±.004
MLS-T	.872±.003	.828±.005	.098±.005
MLS-V	.856±.026	.818±.030	.102±.017
MLS-S	.842±.011	.814±.014	.100±.007

TABLE 4
Comparison among HiMLS Variants on RCV1V2_3 Data Set.

ALGORITHM	F_1-SCORE	ACCURACY	HAMMING LOSS
HIMLS	**.900±.006**	**.869±.006**	**.076±.003**
MLS	.878±.011	.844±.012	.096±.007
MLS-T	.871±.019	.841±.022	.091±.012
MLS-V	.860±.003	.820±.004	.103±.001
MLS-S	.854±.018	.814±.024	.106±.013

TABLE 5
Comparison among HiMLS Variants on RCV1V2_4 Data Set.

ALGORITHM	F_1-SCORE	ACCURACY	HAMMING LOSS
HIMLS	**.894±.002**	**.860±.002**	**.082±.002**
MLS	.874±.010	.835±.012	.097±.008
MLS-T	.864±.019	.825±.023	.103±.014
MLS-V	.859±.016	.816±.020	.106±.013
MLS-S	.851±.013	.807±.016	.113±.011

7.3 Effectiveness of HiMLS Components

First of all, we aim to verify the effectiveness of each component in the proposed model, and demonstrate the advantages of simultaneously modeling the multiple heterogeneity in one framework. Therefore, we compare HiMLS with its four special cases: 1) multi-task multi-view variant MLS; 2) multi-task single-view variant MLS-T; 3) multi-view single-task variant MLS-V; 4) single-task single-view variant MLS-S. Each of these four variants has only one layer.

HiMLS and MLS are input with multi-task and multi-view data. For the single-view setting, the features from all the views are concatenated into one single view. For the single-task setting, the instances in all the tasks are pooled into one single task. For HiMLS, we set the number of layers $L = 2$, and the numbers of latent topics $[p, q]$ for the instances and features(or labels) to $[200, 100]$, $[40, 20]$ in the first and second layer, respectively. For all the other methods with only one layer, we set $[p, q] = [40, 20]$.

The classification performances of HiMLS and its variants on RCV1V2 data sets are shown on Tables 2-5. Based on these comparison results, we have the following findings:

- Both MLS-T and MLS-V perform better than MLS-S in most cases by incorporating either task relatedness or view consistency. It suggests that simply concatenating the features from different views is not the best way to model the view heterogeneity; likewise, simply pooling the instances of all tasks into one single task is not the best way to model the task heterogeneity.
- MLS perform better than either MLS-T or MLS-V in most cases. It suggests that jointly modeling multiple types of heterogeneity can gain performance improvement upon single-heterogeneity learning.
- HiMLS performs better than MLS. It indicates that the learned hierarchical multi-latent space helps build a more robust and discriminative classifier. One possible reason to account for this is that the multi-layer structure helps find the more accurate local optimum by gradually learning the abstract concepts. In contrast, the single-layer methods may suffer from the local optimal solution in lower quality.

7.4 Performance Comparison

The second experiment is to compare the proposed method with various heterogeneous learning algorithms. In this work, we focus on improving the performance of multi-label learning by leveraging the multiple type of heterogeneity. To the best of our knowledge, there is no previous work for learning from the triple heterogeneity. Therefore, we compare our proposed approach with a variety of multi-label learning methods which learn from single or dual heterogeneity. The comparison approaches includes: 1) multi-view multi-label learning methods L^2F [40]; 2) graph-based multi-label approach ML-kNN [45]; 3) multi-label method based on subspace learning LS-ML [22]; 4) transductive multi-label learning approach TRAM [23].

In addition, we compare the two alternative algorithms of our proposed approach, i.e., HiMLS and HiMLSD, to examine their performance differences. Note that HiMLS

TABLE 6
Classification Performance on RCV1V2_1.

ALGORITHM	F_1-SCORE	ACCURACY	HAMMING LOSS
HIMLS	**.906±.006**	**.885±.006**	**.064±.003**
HIMLSD	.889±.007	.858±.007	.082±.003
L^2F	.847±.011	.802±.015	.110±.009
ML-KNN	.803±.092	.775±.094	.102±.038
LS-ML	.821±.021	.789±.019	.109±.005
TRAM	.888±.003	.857±.004	.082±.003

TABLE 7
Classification Performance on RCV1V2_2.

ALGORITHM	F_1-SCORE	ACCURACY	HAMMING LOSS
HIMLS	.903±.006	.872±.006	.073±.003
HIMLSD	**.911±.004**	**.881±.004**	**.071±.002**
L^2F	.884±.005	.850±.005	.083±.003
ML-KNN	.772±.009	.751±.008	.103±.001
LS-ML	.828±.019	.799±.016	.100±.004
TRAM	.874±.004	.848±.004	.079±.002

TABLE 8
Classification Performance on RCV1V2_3.

ALGORITHM	F_1-SCORE	ACCURACY	HAMMING LOSS
HIMLS	**.900±.006**	**.869±.006**	**.076±.003**
HIMLSD	.889±.007	.860±.007	.080±.005
L^2F	.837±.008	.788±.010	.120±.005
ML-KNN	.764±.005	.738±.006	.115±.001
LS-ML	.816±.010	.785±.006	.107±.003
TRAM	.873±.006	.846±.005	.081±.003

TABLE 9
Classification Performance on RCV1V2_4.

ALGORITHM	F_1-SCORE	ACCURACY	HAMMING LOSS
HIMLS	.894±.002	.860±.002	.082±.002
HIMLSD	**.913±.002**	**.866±.004**	.077±.002
L^2F	.858±.005	.816±.005	.106±.005
ML-KNN	.754±.005	.728±.007	.118±.004
LS-ML	.831±.017	.801±.015	.104±.004
TRAM	.870±.004	.851±.006	**.075±.004**

is based on least squares loss function, while HiMLSD is based on generalized KL divergence. In order to conduct a fair comparison between them, the same initializations are used for HiMLS and HiMLSD.

HiMLS (or HiMLSD) is input with multi-task and multi-view data. For the other algorithms, the instances of all the tasks are pooled together. L^2F method is given the multi-view features, whereas the other methods are given the concatenated features from all the views. The parameters are tuned for each algorithm using cross-validation on the training data. We repeat the experiments ten times for each data set and report the average performances and the standard deviations.

Tables 6-9 show the classification performances of d-ifferent methods on RCV1V2. The performances on EUR-Lex, NUS-WIDE, and AL-SMELT are shown in Tables 10-12, respectively.

The results show that both HiMLS and HiMLSD perform better than the other algorithms in most cases. LS-ML [22] learns a common subspace shared among multiple labels, which helps improve the learning performance for the multi-label data. However, since its objective function is non-convex, the performance of LS-ML may be limited by the local optimum problem. TRAM [23] is a transductive multi-label learning method which tries to exploit the information from unlabeled data to estimate the optimal label concept compositions. The results show that unlabeled data can provide helpful information to build the multi-label classifier. For ML-kNN [45], since it ignores the correlation among multiple labels, its performance on these data sets is not comparable with the other methods in most cases. Different from these methods for learning from single heterogeneity, both HiMLS (or HiMLSD) and L^2F [40] model the feature and label heterogeneity and gain performance improvement by enhancing the view consistency. It suggests that treating the features from different views in a discriminative and complementary way is usually better than just concatenating all the features into one view. Likewise, treating the instances in different tasks discriminatively is usually better than just pooling all the instances together. The performance superiority of the proposed method over the comparison methods verifies the effectiveness of the proposed approach to model the complex heterogeneity in a principled framework. Another important competency of the proposed method is that its multi-layer structure helps build a robust classifier by gradually finding the more high-level concepts in the deep structures.

TRAM performs a little better than HiMLS (or HiMLSD) on NUS-WIDE data set. It indicates that NUS-WIDE may be consistent with the smoothness assumption, and TRAM is able to effectively leverage this assumption. However, TRAM shows relative poor performance on AL-SMELT data suggesting that the transductive method may be misled by the unlabeled information.

The results show that the performances of HiMLS and HiMLSD are comparable. Each of them wins on three out of seven data sets. It suggests that they adapt to different data set. Both of them provide the alternative methods to model the heterogeneous data.

TABLE 10
Classification Performance on EUR-Lex.

ALGORITHM	F_1-SCORE	ACCURACY	HAMMING LOSS
HIMLS	**.749±.009**	**.719±.011**	**.033±.002**
HIMLSD	.740±.009	.707±.008	.034±.001
L^2F	.713±.020	.680±.020	.033±.003
ML-KNN	.498±.029	.472±.027	.043±.002
LS-ML	.664±.013	.631±.013	.088±.006
TRAM	.667±.016	.635±.016	.040±.002

TABLE 11
Classification Performance on NUS-WIDE.

ALGORITHM	F_1-SCORE	ACCURACY	HAMMING LOSS
HIMLS	.675±.007	.645±.006	.187±.003
HIMLSD	.649±.008	.634±.009	.192±.005
L^2F	**.700±.001**	.615±.002	.204±.002
ML-KNN	.589±.004	.582±.003	.215±.002
LS-ML	.628±.021	.618±.020	.190±.008
TRAM	.684±.007	**.676±.008**	**.166±.003**

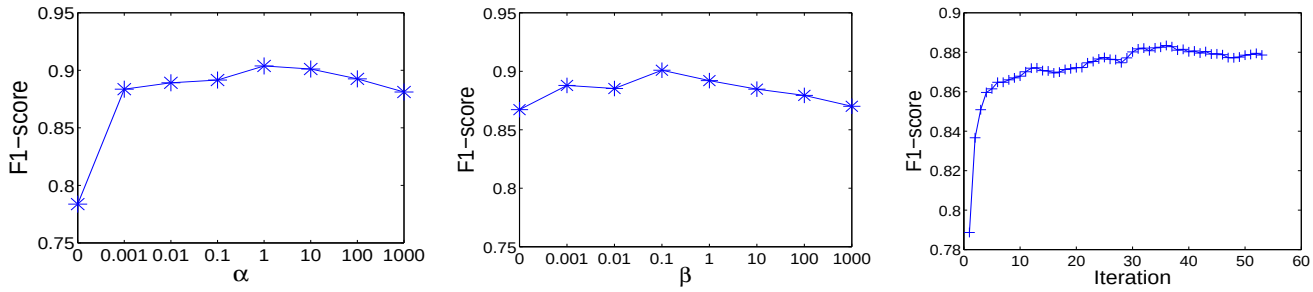


Fig. 2. From left to right: a) F_1-score vs. α (log_{10} scale); b) F_1-score vs. β (log_{10} scale); c) F_1-score vs. iteration.

TABLE 12
Classification Performance on AL-SMELT.

ALGORITHM	F_1-SCORE	ACCURACY	HAMMING LOSS
HIMLS	.848±.003	.847±.003	.140±.003
HIMLSD	**.873±.006**	**.871±.006**	**.115±.006**
L^2F	.773±.001	.772±.001	.214±.000
ML-KNN	.845±.001	.842±.000	.139±.001
LS-ML	.852±.000	.850±.001	.131±.001
TRAM	.383±.001	.381±.001	.605±.000

7.5 Parameter Sensitivity

We study the parameter sensitivity on the RCV1V2_1 data set. α and β are tuned on the grid $10^{[-3:1:3]}$. The results are shown in Figure 2(a-b). α is used to balance the importance of classification loss. The algorithm performs worse as α approaches 0. When $\alpha = 0$, it means that no label information is used for training. The optimal performance is achieved at $\alpha = 1$. Nevertheless, the performance is quite robust over a wide range of values of α. β is used to control the importance of regularization. The result shown in Figure 2(b) indicates that setting appropriate weight to the regularization term can lead to better performance. As a result, we tune the parameters, α and β, for each data set by cross-validation on the training data.

We empirically study the convergence of HiMLS on the RCV1V2_1 data set. The result is shown in Figure 2(c). From this figure, we can see that HiMLS converges fast and its performance becomes stable after a few iterations. Thus, we terminate the algorithm after a maximum of 50 iterations.

7.6 Impact of Layers

TABLE 13
Performance Varies with Number of Layers.

L	F_1-SCORE	ACCURACY	HAMMING LOSS
1	.868±.023	.829±.028	.102±.018
2	.874±.002	.842±.003	.092±.001
3	.884±.003	.855±.005	.083±.002
4	.891±.003	.864±.003	.079±.002
5	**.906±.006**	**.885±.006**	**.064±.003**

It is interesting to investigate how the number of layers L affects the performance of the proposed approach (e.g. HiMLS). We set $L = 1, 2, 3, 4, 5$, and the numbers of latent topics in each layer are 40,100,400,1000,4000, respectively. We set $p = q$ here. Table 13 shows the results on the RCV1V2_1 data set. We can see that the performances (F_1-score, accuracy, and Hamming loss) are consistently improved when the number of layers increased from 1 to 5. It demonstrates that the multi-layer structure improves the performance by learning the hierarchical abstract concepts from data. When L keeps increased from 5, we have not observed the significant improvement of performance. Our conjecture is that the algorithm may have approached the local optimum. Therefore, we empirically set $L = 5$.

8 Conclusion

We propose a multi-layer framework to jointly model triple heterogeneity. In each layer, it learns a multi-latent space shared among the heterogeneous data. Then the multi-latent model is used as a building block to stack up a multi-layer structure so as to gradually learn the more abstract concepts. Based on this generalized framework, we present two alternative models using different divergence measures. A deep learning algorithm is proposed to solve the optimization problem in each model, which first pre-trains each layer and then fine-tunes the whole multi-layer structure by using the multiplicative update rules. The comparison experiments with various heterogeneous learning methods demonstrate the effectiveness of the proposed model.

Acknowledgments

This work is supported by the NSF research grant IIS-1552654, ONR Research grant N00014-15-1-2821, IBM Faculty Award, and NSFC research grant 61473123. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the governments.

References

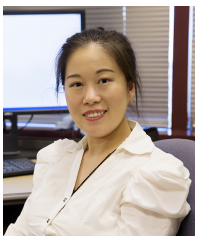
[1] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.
[2] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *NIPS*, pages 41–48, 2006.
[3] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A generalized maximum entropy approach to Bregman co-clustering and matrix approximation. *Journal of Machine Learning Research*, 8:1919–1986, 2007.
[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998.
[5] D. Cai, X. He, J. Han, and T. S. Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1548–1560, 2011.
[6] D. Cai, X. He, X. Wang, H. Bao, and J. Han. Locality preserving nonnegative matrix factorization. In *IJCAI*, pages 1010–1015, 2009.
[7] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.
[8] N. Chen, J. Zhu, and E. P. Xing. Predictive subspace learning for multi-view data: a large margin approach. In *NIPS*, 2010.
[9] T. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. NUS-WIDE: a real-world web image database from national university of singapore. In *CIVR*, 2009.

[10] A. Cichocki, R. Zdunek, A. H. Phan, and S. ichi Amari. *Nonnegative Matrix and Tensor Factorizations - Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley, 2009.
[11] C. H. Q. Ding and X. He. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SDM*, pages 606–610, 2005.
[12] C. H. Q. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(1):45–55, 2010.
[13] C. H. Q. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927, 2008.
[14] C. H. Q. Ding, T. Li, W. Peng, and H. Park. Orthogonal non-negative matrix tri-factorizations for clustering. In *KDD*, pages 126–135, 2006.
[15] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *NIPS*, pages 681–687, 2001.
[16] J. D. R. Farquhar, D. R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmák. Two view learning: SVM-2K, theory and practice. In *NIPS*, 2005.
[17] P. Gong, J. Ye, and C. Zhang. Robust multi-task feature learning. In *KDD*, pages 895–903, 2012.
[18] J. He and R. Lawrence. A graph-based framework for multi-task multi-view learning. In *ICML*, pages 25–32, 2011.
[19] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
[20] S.-J. Huang, Y. Yu, and Z.-H. Zhou. Multi-label hypothesis reuse. In *KDD*, pages 525–533, 2012.
[21] S.-J. Huang and Z.-H. Zhou. Multi-label learning by exploiting label correlations locally. In *AAAI*, pages 1–7, 2012.
[22] S. Ji, L. Tang, S. Yu, and J. Ye. Extracting shared subspace for multi-label classification. In *KDD*, pages 381–389, 2008.
[23] X. Kong, M. K. Ng, and Z.-H. Zhou. Transductive multilabel learning via label set propagation. *IEEE Trans. Knowl. Data Eng.*, pages 704–719, 2013.
[24] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, pages 788–791, 1999.
[25] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.
[26] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.
[27] E. L. Mencía and J. Fürnkranz. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *ECML-PKDD*, pages 126–135, 2008.
[28] A. D. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui. Nonsmooth nonnegative matrix factorization (nsNMF). *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):403–415, 2006.
[29] V. Sindhwani and D. S. Rosenberg. An RKHS for multi-view learning and manifold co-regularization. In *ICML*, pages 976–983, 2008.
[30] K. Sridharan and S. M. Kakade. An information theoretic framework for multi-view learning. In *COLT*, pages 403–414, 2008.
[31] L. Sun, S. Ji, and J. Ye. Hypergraph spectral learning for multi-label classification. In *KDD*, pages 668–676, 2008.
[32] L. Sun, S. Ji, and J. Ye. Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):194–200, 2011.
[33] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. W. Schuller. A deep semi-nmf model for learning hidden representations. In *ICML*, pages 1692–1700, 2014.
[34] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.
[35] D. Wang, T. Li, and C. H. Q. Ding. Weighted feature subset non-negative matrix factorization and its applications to document understanding. In *ICDM*, pages 541–550, 2010.
[36] M. White, Y. Yu, X. Zhang, and D. Schuurmans. Convex multi-view subspace learning. In *NIPS*, pages 1682–1690, 2012.
[37] H. Yang and J. He. Learning with dual heterogeneity: A nonparametric bayes model. In *KDD*, pages 582–590, 2014.
[38] P. Yang and J. He. Model multiple heterogeneity via hierarchical multi-latent space learning. In *KDD*, pages 1375–1384, 2015.
[39] P. Yang, J. He, and J.-Y. Pan. Learning complex rare categories with dual heterogeneity. In *SDM*, pages 523–531, 2015.
[40] P. Yang, J. He, H. Yang, and H. Fu. Learning from label and feature heterogeneity. In *ICDM*, pages 1079–1084, 2014.
[41] H.-F. Yu, P. Jain, P. Kar, and I. S. Dhillon. Large-scale multi-label learning with missing labels. In *ICML*, pages 593–601, 2014.
[42] D. Zhang, J. He, and R. D. Lawrence. MI2LS: multi-instance learning from multiple information sources. In *KDD*, pages 149–157, 2013.
[43] J. Zhang and J. Huan. Inductive multi-task learning with multiple view data. In *KDD*, pages 543–551, 2012.
[44] M.-L. Zhang and K. Zhang. Multi-label learning by exploiting label dependency. In *KDD*, pages 999–1008, 2010.
[45] M.-L. Zhang and Z.-H. Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, pages 2038–2048, 2007.
[46] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.
[47] J. Zhou, J. Chen, and J. Ye. Clustered multi-task learning via alternating structure optimization. In *NIPS*, pages 702–710, 2011.

Pei Yang is a postdoctoral researcher in the Statistical Learning Lab (STAR Lab), Arizona State University. His research focuses on statistical machine learning and data mining such as heterogeneous learning, semi-supervised learning, transfer learning, rare category analysis, and functional data analysis, with applications in web mining, big data analytics, bioinformatics, healthcare, etc. He has published over 20 research articles on referred journals and conference proceedings.

Hasan Davulcu is an associate professor in the School of Computing, Informatics and Decision Systems Engineering at Arizona State University. He has done research in data mining and information assurance. His previous works in data and services integration were published at prestigious ACM and IEEE conferences. He is currently the PI for an NSF Partnership for Innovation: Building Innovation Capacity (PFI:BIC) grant focusing on financial fraud detection via visual analytics. Dr. Davulcu holds a Ph.D. in computer science from the State University of New York at Stony Brook, New York.

Yada Zhu is a research staff member from IBM T. J. Watson Research Center. Her research interests include big data analytics, survival analysis, statistical data mining and machine learning with applications to ecommerce, advanced manufacturing, energy and utilities. Dr. Zhu has published over 30 research articles on referred journals, books and conference proceedings. Her work has been acknowledged by IBM innovation awards and IBM research accomplishment awards. Dr. Zhu has served as an Associated Editor of International Journal QTQM.



Jingrui He is an assistant professor in the School of Computing, Informatics and Decision Systems Engineering at Arizona State University. She received her PhD in Computer Science from Carnegie Mellon University. She joined ASU in 2014 and directs the Statistical Learning Lab (STAR Lab). Her research focuses on heterogeneous machine learning, rare category analysis, semi-supervised learning and active learning, with applications in healthcare, social network analysis, semiconductor manufacturing, etc. She is the recipient of the NSF CAREER Award in 2016, IBM Faculty Award in 2015 and 2014 respectively, and has published more than 60 refereed articles. She has served on the organizing committee/senior program committee of many conferences, including ICML, KDD, IJCAI, SDM, ICDM, etc. She is also the author of the book on Analysis of Rare Categories (Springer-Verlag, 2012).