Deep Multi-type Objects Muli-view Multi-instance Multi-label Learning*

Yuanlin Yang†, Guoxian Yu†‡, Carlotta Domeniconi§, Xiangliang Zhang¶

Abstract

Multi-view multi-instance multi-label learning (M3L) can model complex objects (bags) that are composed of multiple instances, represented with heterogeneous feature views and annotated with multiple related semantic labels. Although significant progress has been made toward M3L tasks, the current solutions still focus on a single-type of complex objects, and cannot effectively mine the widely-witnessed interconnected objects of multi-types. To bridge this gap, we propose a Deep Multi-type objects Multi-view Multi-instance Multi-label Learning solution (DeepM4L) based on heterogeneous network embedding. DeepM4L first encodes the inter- and intra-relations among multi-type objects using a heterogeneous network, and performs instance neighbor embedding to learn the representation vectors of instances. Next, it obtains the instance-label score tensor for each view and uses a max pooling operation to induce the bag-label score tensor for each bag. After that, it combines bag-label scores by multi-view learning to guarantee the semantic consistency between bags of different views. Our empirical study on benchmark datasets shows that DeepM4L is significantly superior to the recent advanced baselines.

1 Introduction

Diverse real-world data (including images and texts) are usually associated with multiple semantic labels and represented with heterogeneous feature views, which describe the complex object from different aspects. Taking Fig. 1 for example, the article object (bag) contains multiple instances (paragraphs and patches) from text view and image view, respectively. This type of multi-view multi-instance objects are also simultaneously tagged with multiple semantic related labels (or topics). Multi-view Multi-instance Multi-label Learning (M3L) has been developed to deal with such complex data [1, 2, 3]. M3L aims to leverage the relationships between instances, bags, semantic labels and between heterogeneous feature views to predict the labels of objects and those of individual instances affiliated with these objects. However, M3L still focuses on *single-type* objects. Considering the diverse interconnections between multiple types of objects, the labels of complex objects are determined not only by their own feature views, but also by their connections with other types of objects. Therefore, M3L lacks the ability to jointly model multi-types of objects.

One bypass solution is to form the additional feature vectors of target objects by projecting other types of objects toward the target objects using the interconnections. This projection has been extensively applied in multi-view learning [4, 5] and multiple kernel learning [6]. But how to make such projection without corrupting the intrinsic structure and attribution information of these objects is an open problem [7]. Although some matrix factorization based solutions [8, 9, 10, 11] and heterogeneous network embedding approaches [12, 13, 14] can mine the structure and attribute information of interconnected objects, these methods still ideally consider each object indivisible. In practice, however, these objects are further made of different instances, which correspond to different salient parts (i.e., title, abstract, keywords of an article) of this object. How to mine these interconnected complex objects of different types, to the best of our knowledge, is still not well studied yet.

To learn from multi-type interconnected complex objects, a new learning paradigm termed as Multi-type objects Multi-view Multi-instance Multi-label Learning (M4L) is recently proposed and a joint matrix factorization based solution (M4L-JMF) is introduced to model this new learning paradigm [15]. M4L-JMF firstly uses multiple data matrices to separately store the attributes and multiple inter(intra)-associations among bags, and then jointly factorizes these matrices into low-rank ones to explore the latent representation of each bag and its instances. M4L-JMF further uses a dispatch and aggregation objective to dispatch the labels of bags to individual instances and reversely aggregate the labels of instances to the hosting bags in a coherent manner.

*Supported by NSFC (61872300, 62031003 and 62072380), Corresponding Author: Guoxian Yu (guoxian85@gmail.com).

†College of Computer and Information Sciences, Southwest University, yyl110000@email.swu.edu.cn.

‡School of Software, Shandong University, guoxian85@gmail.com

§George Mason University, carlotta@cs.gmu.edu.

¶King Abdullah University of Science and Technology, xiangliang.zhang@kaust.edu.sa

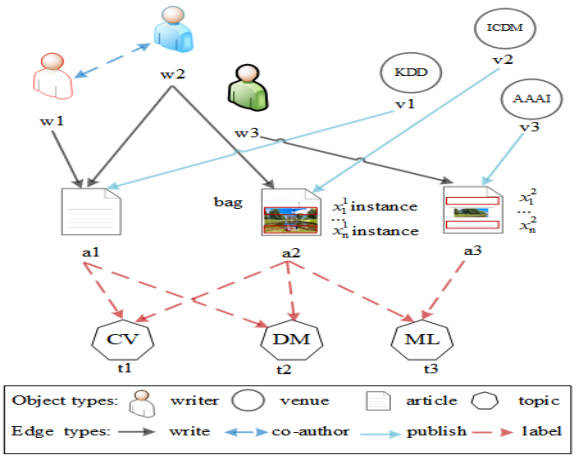


Figure 1: An illustration of Multi-type Multi-view Multi-instance Multi-label (M4) data of four types (writer, venue, article and topic). The article object (bag) is represented with both image and text views, further composed of various instances (i.e., paragraph x_1^2 in $a3$ and image patch x_1^1 in $a2$), and associated with multiple semantic labels (i.e., CV and DM). These inter-connected multi-type objects naturally form a heterogeneous network.

However, M4L-JMF only learns a linear representation of these objects and can not scale up to large scale data, due to the high complexity of matrix factorization.

We introduce a **Deep** M4L (**DeepM4L**) solution. DeepM4L constructs a heterogeneous information network ($\mathcal{R}$) made of nodes of objects (bags), instances, and labels, to uniformly encode multiple inter-relations between multi-type objects, the associations between instances and their affiliated bags, the semantic relations between bags and labels, and the correlations between labels. Inspired by the expressive power of heterogeneous information network for modeling interconnected nodes of different types, it is straightforward to build DeepM4L based on heterogeneous network embedding (HNE) techniques [16, 12, 17, 14]. Compared with the heavily studied homologous network embedding (NE), HNE can accommodate not only multi-types of objects, but can also merge all types of connections to learn comprehensive representation vectors of objects for diverse tasks, such as link prediction and node classification [18]. Although the current HNE solutions have achieved encouraging results in diverse tasks, it still faces with three challenges for M4L task. **Challenge 1**: HNE and NE solutions consider complex objects as simple network nodes but ignore the instance-level subobjects, which build up the bag and contain important context and semantic information for the hosting bag. Therefore, current HNE and NE solutions are difficult to directly apply for multi-instance data. **Challenge 2**: The same bag across views can have varying instances and how to ensure the label consistency between a bag and its affiliated instances within/between views is a non-trivial job. **Challenge 3**: The previous matrix factorization methods [3, 19, 15] can only learn a linear representation with respect to a small number of objects, they disregard the complex nonlinear relations and thus have a limited representation ability.

To address these challenges of applying HNE for mining M4 data, we first construct an attributed heterogeneous network to encode the inter- and intra-relations among multi-type bags, and then apply an instance embedding network to learn the composite representation feature vectors of instances and bags embedded in the HNE space (For Challenge 1 and 3). Next, to make the label consistency between instance-level and bag-level within and between views, we use a multi-instance pooling layer to generate the bag label score tensor of each bag across views, and finally apply multi-view learning to fuse the label score tensors of the multi-view bags to predict bag-level and instance-level labels in a coherent fashion (For Challenge 2).

The main contributions of this work are:
(i) DeepM4L encodes multi-types of objects and their affiliated instances in a heterogeneous information network, and performs instance neighborhood embedding to learn complex non-linear representation, and thus it can efficiently mine both the attribute and local network structure information of interconnected multi-type objects.
(ii) DeepM4L not only guarantees the label consistency between the bag and instance levels; it also pursues the label consistency of bags across views by fusing multi-view bag-label score tensors.
(iii) Experimental results on benchmark datasets show that DeepM4L outperforms the representative M3L solutions (M3Lcmf [3], M2IL [20] and ICM2L [21]), data fusion solutions (MFDF [8] , SelDFMF [10] and M4L-JMF[15]) based on matrix factorization, and network embedding methods (metapath2vec [22] and GraphHeat [23]) for diverse M4L tasks.

The rest of the paper is organized as follows. Section 2 reviews the related work and Section 3 elaborates on the details of the proposed Deep M4L. The experimental results and analysis are presented in Section 4, and Section 5 concludes the paper.

2 Related work

Our work has close connections with M3L and network embedding, but it is a more general framework than M3L and its degenerated variants (Multi-view Multi-

label Learning [24], Multi-view Multi-instance Learning [20], and Multi-instance Multi-label Learning [25]. Note, multi-instance learning can also viewed as a special case of multi-relational learning [26]). These learning paradigms consider only one type of objects, each of which is composed of multiple instances, or look at instance information from multiple views. M3L has achieved much progress during the past decades. To the best of author knowledge, Multi-instance Multi-label Latent Dirichlet Allocation (M3LDA) [1] is the first M3L algorithm, it learns a visual-label part from the visual view and a text-label part from the text view, and forces labels decided by the textual and visual information being consistent to label complex objects. Multi-Instance Multi-Label mixture (MIMLmix) [27] uses a hierarchical Bayesian network and mixture topic model to label complex objects. Hierarchical Music Emotion Recognition model (HMER) [28] captures music emotion dynamics with a song-segment-sentence hierarchical structure, and considers emotion correlations between music segments and sentences to annotate M3 data. Multi-modal Multi-instance Multi-label Deep Network (M3DN) [2] learns the label prediction and exploits label correlation using the Optimal Transport, and the consistency principle between different modal bag-level prediction and the learned latent ground label metric. M3L based on collaborative matrix factorization (M3Lcmf) [3] utilizes a heterogeneous network to encode different types of relations between bags, instances and labels, and then collaboratively factorizes the relational data matrices into low-rank matrices to seek the latent representations of bags, instances and labels. M3Lcmf finally reconstructs the bag-label/instance-label relations by matrix completion. M3DNS [29] considers the instance-level auto-encoder for single modality and the modified bag-level optimal transport to strengthen the consistency among modalities, and leverages the instance-level and bag-level information to predict the labels of bags. Weakly-supervised M3L (WSM3L) [19] studies M3L in a more general setting with unpaired view data and missing labels. Particularly, WSM3L adapts multi-modal dictionary learning to learn a shared dictionary (representational space) across views and individual encoding vectors of bags for each view to seek the match between bags across views, and to coherently predict the labels of bags and instances. These M3L methods simply consider only one type of complex objects. Although we can merge the feature information from other interconnected objects and then apply M3L methods, but this merge is typically man-made and may distort the intrinsic structures among these interconnected objects [8, 9].

Some recent matrix factorization and network embedding based solutions [8, 10, 18] can integrate interconnected multi-type of objects, but they still neglect the fact that complex objects can be further made of various instances (i.e., patches of an image, and paragraphs of a text). Compared with matrix factorization based solutions, which are limited to moderate and Euclidean data, network embedding-based solutions have shown its potential of mining nonlinear non-Euclidean graph data at a larger scale, successfully applied in many data mining tasks (such as node classification, link prediction, recommendation system) [30]. Unlike plain NE-based solutions [31, 31, 32], HNE-based solutions can accommodate diverse types of objects and interconnections [18].To name a few, [33] introduces a HNE solution to capture the complex interactions between heterogeneous nodes by highly nonlinear multi-layered embedding function. Metapath2vec[22] defines the neighbor of nodes via meta-path and learns the node embedding by skip-gram with negative sampling. Heterogeneous network embedding by Generative Adversarial Networks (HeGAN) [13] designs a relation-aware discriminator and generator to learn node embedding based on the adversarial principle. Attributed Multiplex Heterogeneous Network embedding (AMHEN) [12] further considers attributed heterogeneous network by edge type aware neighborhood message passing and aggregating. However, these solutions still consider simple network nodes, and neglect the fact that complex nodes can be further made of multiple instances, which cause the loss of more granular information, such as the key paragraphs and salient image patches of the article node in Fig. 1.

To capture the fine-grained information at the instance-level and join the power of heterogeneous information networks for encoding multiplex nodes, we introduce a HNE based approach called DeepM4L to comprehensively model interconnected complex objects of different types. The following section elaborates on its procedure.

3 The Proposed Method

3.1 Problem Statement A heterogeneous information network $\mathcal{G} = \{\mathcal{V}, \mathcal{R}\}$ is a special kind of network, which contains multiple types of objects $\mathcal{V}$ and/or multiple types of links $\mathcal{R}$. Unlike a typical heterogeneous information network (HIN), that is composed of multiple objects/relations of different types, the HIN for M4L contains complex bag types, which are further made of diverse instances. Suppose there are m types of interconnected objects, $\mathbf{X}_{bi}^v = \{\mathbf{x}_{i1}^v, \mathbf{x}_{i2}^v, \cdots, \mathbf{x}_{in_i^v}\}$, where $\mathbf{x}_{ij}^v \in \mathbb{R}^d$ is the feature space of instances (if any) for the i-th object, and n_i^v is the number of instances for this bag in the v-th view. $\mathbf{Y} = [\mathbf{y}_1; \mathbf{y}_2; \cdots; \mathbf{y}_n]$ denotes

Table 1: Main notations used in the paper.

Notation	Explanation
$\mathcal{G}$	Heterogeneous information network
$\mathcal{V}$	Multiple types of objects
$\mathcal{R}$	Multiple types of links
$\mathbf{X}_{bi}^{v}$	Feature space of instances for i-th bag
$\mathbf{Y}$	Label vector of bags
$\mathbf{R}_{ij}$	Inter-relational data matrices
$\mathbf{A}$	Attribute data matrices of objects
$\mathbf{G}_{i}^{v}$	Instance-label score tensor
$\mathbf{R}_{ib}^{v}$	Instance-bag association tensor
$\mathbf{R}_{bl}^{v}$	Bag-label score tensor for the v-th view

the label vectors of n bags, $\mathbf{y}_i \in \mathbb{R}^q$ is the q-dimensional one-hot label vector for the i-th bag, where $\mathbf{y}_{ic} = 1$ means that the i-th bag has the c-th positive label, and $\mathbf{y}_{ic} = 0$ otherwise. The main symbols used in this paper are given in Table 1.

DeepM4L aims to learn a mapping function $f(\mathcal{V}, \mathcal{R}, \mathcal{A}) \in \{0,1\}^q$ to annotate the target objects (i.e., papers) with respect to q distinct but related labels (i.e., DM, CV and ML). Here, $\mathcal{R}$ collectively stores all the inter-relational data matrices $\mathbf{R}_{ij}$ for the i- and j-th types of objects, $\mathcal{A}$ collectively stores all the attribute data matrices $\mathbf{A}$ of objects.

3.2 Instance Neighbor Embedding The feature representation of instances affects the performance of multi-instance multi-label learning [34]. Besides the representation of a bag induced from its hosted instances, we also need to fuse the feature/structure information of other types of objects. Here we use the heterogeneous network schema to encode the structural relations of multi-types of objects. We first use the heterogeneous network as the input and use the embedding model to get the k-dimensional vector $\mathbf{z} \in \mathbb{R}^k$ of each instance node. Next, we consider the first-order neighbors of a node to gather and disseminate information. Then, we consider different inter- or intra-nodes to obtain *one*-order neighbors as follows:

$$\mathbf{h}_i^{(c+1,v)} = \varphi\Big(\frac{1}{p_r^i}\mathbf{h}_i^{(c,v)} + \sum_{\mathrm{r}\in\mathcal{R}}\sum_{j\in\mathcal{N}_r^i}\frac{1}{p_r^{ij}}\mathbf{W}_r^{(c,v)}\mathbf{h}_j^{(c,v)}\Big) \tag{3.1}$$

where $\mathcal{N}_r^i$ denotes the set of neighbor indices of node i for relation type r, p_r^{ij} and p_r^i are normalization constants that can chosen in advance (e.g. $p_r^{ij} = \sqrt{(|\mathcal{N}_r^i||\mathcal{N}_r^j|)}$ and $p_r^i = |\mathcal{N}_r^i|$). $\mathbf{h}_i^{(c,v)} \in \mathbb{R}^k$ is the hidden state of node i in the c-th layer of neural network for the v-th view, and the embedded aggregation information capability of a node is determined by C (number of layers). $\mathbf{W}_r^{(c,v)}$ is a relation-type specific parameter matrix for the c-th layer and the v-th view, φ is an activation function, such as rectified linear unit. We can get a deep model through chaining multiple layers, thus learning the multi-level representation of node v_i and getting the final embedding $\bar{\mathbf{x}}_i^v = \mathbf{h}_i^{(C,v)}$, $\bar{\mathbf{x}}_i \in \mathbb{R}^k$. Fig. 2(i) illustrates the $c+1$ layer operation. The entire embedding adopts the following form. We stack C layers according to the definition in Eq. (3.1), so that the output of this convolutional layer becomes the input of the next hidden layer. The input of the first layer is the instance feature vector $\mathbf{h}_i^{(0,v)} = \mathbf{x}_i^v$, an adjacency list vector or one hot vector is provided for other types of nodes in the networks. Our goal is to get the embedded representation of instance $\bar{\mathbf{x}}_i^v = \mathbf{h}_i^{(C,v)} \in \mathbb{R}^k$ by fusing the connections with other types of objects and attribute information. By doing so, we get the i-th bag $\bar{\mathbf{X}}_i^v = \{\bar{\mathbf{x}}_{i1}^v, \cdots, \bar{\mathbf{x}}_{in_i^v}^v\}$ with n_i^v instances. Finally, we use the full connection layer and softmax to get the instance-label score tensor $\mathbf{G}_i^v$, the process can be formulated as follows:

$$\mathbf{G}_i^v = f_\Omega(\bar{\mathbf{X}}_i^v) \tag{3.2}$$

where $f_\Omega(\bar{\mathbf{X}}^v)$ stands for a network generating instance-label score tensor, and $\mathbf{G}_i^v \in \mathbb{R}^{n_i^v \times q}$.

3.3 Objective Function Since the label of a bag is reflected by its instances, we aggregate the obtained labels of instances to the hosting bag through the instance-score tensor column max pooling (as shown Eq. (3.3)). Fig. 2 (ii) shows the process for the i-th bag $\bar{\mathbf{X}}_i^v$. We can then obtain the multi-view bag-label score tensor, which is meaningful for the multi-view multi-instance setting, since complex objects generally have the bag-level labels but miss the instance-level ones. The bag-level label tensor generation process can be formulated as follows:

$$\mathbf{R}_{bl}^v = g_\Theta(f_\Omega(\bar{\mathbf{X}}^v)) \tag{3.3}$$

where $g_\Theta(f_\Omega(\bar{\mathbf{X}}^v))$ represents the bag-label mapping network based on the input of instance embedding $f_\Omega(\bar{\mathbf{X}}^v)$, and $\mathbf{R}_{bl}^v \in \mathbb{R}^{n\times q}$. Through Eq. (3.3), we can get the bag label relation tensor for the v-th view. For example, $\mathbf{R}_{bl}^1 \in \mathbb{R}^{n\times q}$ and $\mathbf{R}_{bl}^2 \in \mathbb{R}^{n\times q}$ are bag-label tensors on the text or image view, respectively.

With the instance neighbor embedding, DeepM4L gets the representation of instances and their bags $\bar{\mathbf{X}}^v$, which not only learns instance correlation, but also obtains the multi-typed object information in HIN. After getting the bag label score tensor by max pooling the multi-instance label tensor for each view, we need to unify these tensors to predict the labels of bags and those of instances as follows:

$$\mathbf{Y} = g_\Theta(f_\Omega(\bar{\mathbf{X}}^v)) + \mathbf{E}^v \tag{3.4}$$

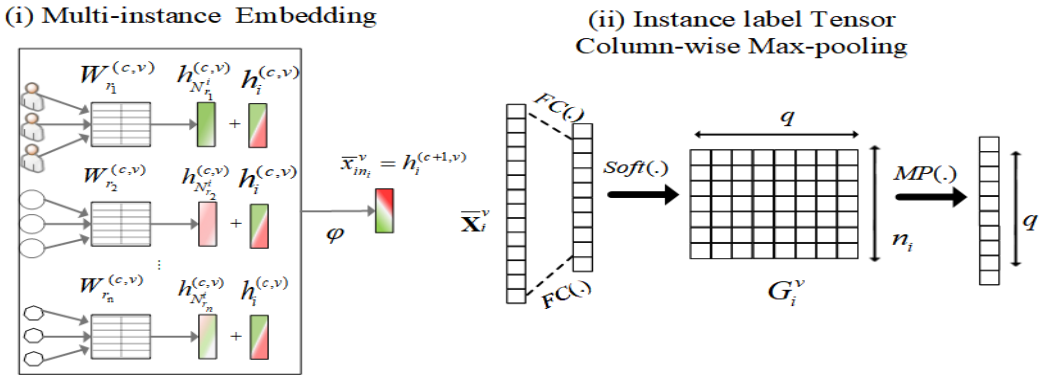


Figure 2: An illustration of DeepM4L. (i) shows the C layer instance embedding in the view v, and learns the network structure information of the author, venue, and topic. (ii) Shows the column max pooling operation for the instance label 2D tensor $\mathbf{G}_i^v$, $\bar{\mathbf{X}}_i^v = \{\bar{\mathbf{x}}_{i1}^v, \bar{\mathbf{x}}_{i2}^v, \cdots, \bar{\mathbf{x}}_{in_i}^v\}$ is the i-th bag final embedding representation from (i). $FC(\cdot)$ means the full connection operation, $Soft(\cdot)$ means softmax operation, and $MP(\cdot)$ is the column-wise max-pooling operation.

where $\mathbf{E}^v \in \mathbb{R}^{n\times q}$ denotes the view-dependent reconstruction error for the v-th view. $\mathbf{Y} \in \mathbb{R}^{n\times q}$ is the bag-label ground truth across all views. $\mathbf{Y}$ is not only used as supervisory information to guide network training and learning, but also to align view-wise tensors along with the known labels of bags, and thus push predicted bag-level labels to individual instances. Finally, we give the objective function of DeepM4L as:

(3.5)

$$\min_{\Omega,\Theta} \mathcal{L}(\{\bar{\mathbf{X}}^v\}_{v=1}^V, \mathbf{Y}) = \sum_{v=1}^{V} ||f_\Omega(\bar{\mathbf{X}}^v) - \mathbf{R}_{ib}^v g_\Theta(f_\Omega(\bar{\mathbf{X}}^v))||_F^2 + \sum_{v=1}^{V} ||g_\Theta(f_\Omega(\bar{\mathbf{X}}^v)) - \mathbf{Y}||_F^2$$

$||f_\Omega(\bar{\mathbf{X}}^v) - \mathbf{R}_{ib}^v g_\Theta(f_\Omega(\bar{\mathbf{X}}^v))||_F^2$ is added to calculate the instance level label prediction loss. $||g_\Theta(f_\Omega(\bar{\mathbf{X}}^v)) - \mathbf{Y}||_F^2$ aims to integrate bag-label tensors across views. DeepM4L has two objectives: the first one is to calculate the instance-label relation loss for each view, and the back-propagation of loss can allow instance neighbor embedding to gradually improve its embedding ability; the second one is to integrate bag-label tensors across views to reach the consensus bag-label score. The former is pursued by instance neighbor embedding per view; the purpose of the latter is to align the predicted multi-view bag-label tensor scores with the known bag-label score. They are unified in Eq. (3.5).

From the above analysis, we can conclude that DeepM4L can predict labels for complicated objects both at instance level and bag level, and can simultaneously preserve diverse relations among multi-types of nodes. In addition, DeepM4L can learn complex non-linear representations and mine complex relationships of multi-type objects.

4 Empirical Evaluation

4.1 Experimental Setup We designed three experiments to study the performance of our proposed algorithm. In the first experiment, we used real world M4 data (Isoform dataset) to study the performance of DeepM4L and compare it against M3L and network embedding methods. In the second experiment, we compared DeepM4L against M3L and its degenerated versions on benchmark multi-instance multi-label datasets [35, 36, 37, 3]. The third experiment explores the performance of DeepM4L and the classical graph embedding methods on the LncRNA dataset [9]. We compare the performance of DeepM4L against some related algorithms, including M3L solutions (M3Lcmf [3], M2IL[20], ICM2L [21]), data fusion solutions based on matrix factorization (DFMF [8], SelMFDF [10]), network embedding (metapath2vec [22], GraphHeat [23]). The input parameters of these comparison methods are specified (or optimized) according to the recommendations of the authors in their codes or papers. As to the proposed DeepM4L, we set the maximum number of epochs as 200, and layer number from $\{1, 2, \cdots, 10\}$. We use the Adam optimizer to train the model for up to 200 epochs (training iterations) [38] with a learning rate of 0.005. We initialize the weights based on a uniform distribution and the size of the previous layer [39]. If the verification loss does not decrease for two consecutive epochs, the training will stop.

The statistical information of these datasets is listed in Table 2. **Isoform** is composed of 5 types of

Table 2: Statistics of datasets used for the experiments. *avgBL* is the average number of labels per bag and *avgBI* is the average number of instances per bag.

Dataset	Bags	Instances	Labels	AvgBL	AvgBI	View	Node Type	Link Type
Letter Frost	144	565	26	3.6	3.9	1		
Letter Carroll	166	717	26	3.9	4.3	1		
MSRC v2	591	1,758	23	2.5	1.0	1		
Birds	548	10,232	13	2.1	18.7	1		
Isoform	8,000	76,244	6,428	16.9	6.5	2	5	5
LncRNA	240		412			6	6	9

objects (miRNAs(495), genes(8,000), isoforms(76,244), Gene Ontology(6,428), Disease Ontology(8,450)) and 5 link types, the more detailed information can be found in [40]. **LncRNA** datasets are composed of 6 types of objects (LncRNA(240), miRNAs(495), genes(15,527), Gene Ontology(6,428), Disease Ontology(412) and Drug(8,283)) and 9 correlation types. These two heterogeneous biological datasets were used to predict the associations between isoforms/lncRNAs and functions/diseases. Additional information can be found in [10, 41]. The other four single-view multi-instance multi-label datasets have been extensively used in multi-instance multi-label learning [37, 3, 15].

To effectively evaluate the performance of DeepM4L, we adopted three frequently used evaluation metrics in multi-label learning and bioinformatics, namely the average area under the precision-recall curve (*AUPRC*), the average area under the receiver operating curve (*AUROC*), and the average F1-score (*AvgF1*) across all classes. The formal definitions of these metrics are omitted due to the page limit, and can be found in [42].

4.2 Results on Isoform We randomly partition the samples of each dataset into a training set (70%) and the remaining are for test, and independently run each algorithm in each partition. We report the average results (10 random partitions) and standard deviations for each experiment. To study the performance of DeepM4L, we apply it on the Isoform dataset to predict the associations between isoforms and Gene Ontology terms (the functional labels of isoforms). Because the M3 methods cannot directly handle multiple types of objects, we first project other objects towards the genes to form M3 data, and then apply M3L methods. For matrix factorization based data fusion (except M4L-JMF) and heterogeneous network embedding methods, we adopt its classic setting [15], ignoring the specific bag-instance associations and combine multiple types of objects for the prediction task. We use the top K labels corresponding to the largest entry in each row of $\mathbf{R}_{bl}$ ($\mathbf{R}_{il}$) as the relevant labels of the bag (instance). Here K is the average number of labels per bag/instance.

We report the results of the first experiment in Table 3. We have two interesting observations. The first is that the instance neighbour embedding not only can effectively fuse neighbor information of the same type, but also explore information from other types of objects, and thus significantly increase AvgF1 by at least 2%. Compared with M4L-JMF, the AvgF1 value increases more obviously. This is contributed by the non-linear representation learned by DeepM4L. In contrast, M4L-JMF only learns linear representation by joint matrix factorization. DeepM4L loses to some compared methods with respect to AUROC, the possible cause is that the adopted datasets are imbalanced, and AUROC is less sensitive to imbalanced data than AvgF1 and AUPRC. Another interesting observation is that the fusion of other types of objects can significantly improve the bag-label relationship prediction, which comes from the comparison with M3L and other methods that incorporate multiple types of objects. The AUROC and AUPRC values of M3L and M4L methods are not only high, but also more stable than single-type objects learning based methods. This fact suggests that there is valuable information hidden in the HIN.

Table 3: Results on Isoform of DeepM4L and related methods. •/○ indicates whether DeepM4L is statistically (according to pairwise t-test at 95% significance level) superior/inferior to the other method.

Method	**AvgF1**	**AUROC**	**AUPRC**
M3Lcmf	0.174±0.006•	0.675±0.013•	0.162±0.015•
ICM2L	0.079±0.002•	0.555±0.001•	0.056±0.019•
M2IL	0.037±0.005•	0.561±0.006•	0.055±0.014•
DFMF	0.055±0.002•	0.948±0.007○	0.641±0.039•
SelMFDF	0.049±0.001•	0.951±0.003○	0.646±0.023•
Metapath2vec	0.231±0.001•	0.741±0.001•	0.452±0.037•
M4L-JMF	0.057±0.003•	0.969±0.002○	0.676±0.021•
GraphHeat	0.291±0.013•	0.865±0.002•	0.358±0.017•
DeepM4L	0.315±0.013	0.873±0.003	0.697±0.015

M3L methods and its degenerated versions (ICM2L and M2IL) have a lower performance than M4L solutions and network embedding based solutions. That is because the former methods only learn the shallow representation of objects and consider single-type of objects. M3Lcmf uses a heterogeneous network to capture different types of inter(intra)-relational data, and then collaboratively factorizes the relational data matrices of

Table 4: Results on four datasets with bag-level labels of DeepM4L and related methods. •/○ indicates whether DeepM4L is statistically (according to pairwise t-test at 95% significance level) superior/inferior to the other method.

Metric	ICM2L	M2IL	M3Lcmf	DFMF	SelMFDF	M4L-JMF	DeepM4L
				Birds			
AvgF1	0.372±0.009•	0.023±0.007•	0.485±0.012•	0.252±0.012•	0.261±0.009•	0.268±0.014•	0.498±0.007
AUROC	0.720±0.008•	0.541±0.008•	0.604±0.032•	0.902±0.009	0.912±0.003○	0.944±0.002○	0.897±0.001
AUPRC	0.255±0.019•	0.014±0.006•	0.425±0.012•	0.886±0.045•	0.891±0.018•	0.963±0.008•	0.971±0.007
				Letter Carroll			
AvgF1	0.317±0.001•	0.047±0.007•	0.543±0.044•	0.247±0.014•	0.269±0.021•	0.288±0.014•	0.558±0.009
AUROC	0.589±0.006•	0.512±0.013•	0.649±0.022•	0.906±0.008○	0.921±0.004○	0.924±0.002○	0.897±0.003
AUPRC	0.101±0.002•	0.013±0.006•	0.421±0.012•	0.909±0.056•	0.913±0.027•	0.948±0.008	0.952±0.003
				Letter Frost			
AvgF1	0.252±0.001•	0.063±0.007•	0.538±0.050•	0.242±0.018•	0.246±0.029•	0.250±0.009•	0.576±0.023
AUROC	0.638±0.010•	0.531±0.026•	0.665±0.002•	0.907±0.003•	0.912±0.004○	0.924±0.002○	0.909±0.002
AUPRC	0.211±0.024•	0.020±0.002•	0.495±0.083•	0.894±0.049•	0.895±0.029•	0.951±0.008	0.957±0.012
				MSRC v2			
AvgF1	0.219±0.003•	0.024±0.007•	0.426±0.028•	0.198±0.008•	0.213±0.007•	0.215±0.004•	0.449±0.013
AUROC	0.668±0.063•	0.573±0.053•	0.698±0.018•	0.937±0.002○	0.939±0.001○	0.958±0.001○	0.917±0.002
AUPRC	0.253±0.026•	0.057±0.006•	0.444±0.083•	0.880±0.021•	0.905±0.009•	0.933±0.003•	0.941±0.005

Table 5: Results on four datasets with instance-level labels of DeepM4L and related methods. •/○ indicates whether DeepM4L is statistically (according to pairwise t-test at 95% significance level) superior/inferior to the other method.

Metric	M3Lcmf	M4L-JMF	DeepM4L
		Birds	
AvgF1	0.286±0.000•	0.291±0.017•	0.512±0.011
AUROC	0.515±0.003•	0.957±0.018○	0.921±0.008
AUPRC	0.445±0.014•	0.961±0.004•	0.968±0.003
		Letter Carroll	
AvgF1	0.104±0.012•	0.085±0.005•	0.231±0.018
AUROC	0.522±0.015•	0.891±0.031○	0.847±0.022
AUPRC	0.367±0.011•	0.913±0.014•	0.928±0.009
		Letter Frost	
AvgF1	0.352±0.107	0.072±0.005•	0.371±0.011
AUROC	0.517±0.016•	0.894±0.011○	0.851±0.009
AUPRC	0.472±0.017	0.884±0.009	0.897±0.012
		MSRC v2	
AvgF1	0.208±0.074•	0.147±0.002•	0.293±0.003
AUROC	0.553±0.009•	0.885±0.013○	0.839±0.015
AUPRC	0.458±0.007•	0.867±0.021	0.873±0.017

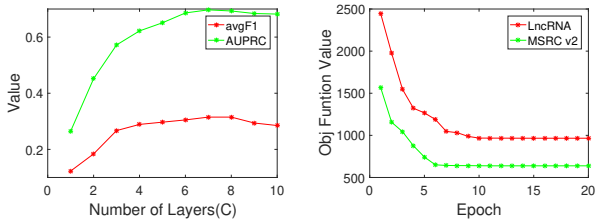
Table 6: Results of DeepM4L, matrix factorization based and network embedding methods on the LncRNA dataset. •/○ indicates whether DeepM4L is statistically (according to pairwise t-test at 95% significance level) superior/inferior to the other method.

Method	AvgF1	AUROC	AUPRC
DFMF	0.062±0.001•	0.872±0.007○	0.546±0.091•
SelMFDF	0.066±0.003•	0.887±0.003○	0.604±0.015•
M4L-JMF	0.067±0.002•	0.895±0.004○	0.616±0.026
metapath2vec	0.096±0.002•	0.771±0.013•	0.375±0.014•
GraphHeat	0.128±0.001•	0.836±0.019○	0.492±0.006•
DeepM4L	0.175±0.001	0.842±0.014	0.626±0.011

the network into low-rank matrices to explore the potential relationships between bags, instances and labels among multiple views. As a result, it often works better than the other two degenerated M3L methods (M2IL and ICM2L). metapath2vec performs meta-path-based random walks to construct the heterogeneous neighborhood of a node and then leverages a heterogeneous skip-gram model to perform node embeddings. However, due to the limitation of sampling capacity and the lack of consideration of bag-instance association, it beats by DeepM4L. GraphHeat leverages the local structure of target node under heat diffusion to flexibly determine its neighboring node, but it does not consider the valuable bag-instance relation also, and thus has a compromised performance. M4L-JMF takes into account bag-instance associations, its performance is significantly better than other matrix factorization based data fusion methods (DFMF and SelMFDF) and even the network embedding methods. However, since M4L-JMF can only learn a linear instance/bag representation of complex objects, its AvgF1 and AUPRC values are lower than our proposed DeepM4L, which not only models bag-instance associations via a pooling layer and by aligning the original bag-label score, but also mines nonlinear representations of bags and instances by network embedding.

Overall, the results on Isoform suggest the effectiveness of DeepM4L on modeling bag-instance associations and learning nonlinear representation of bags and instances. These results also confirm the rationality of DeepM4L on unifying multi-instance learning with heterogeneous network embedding.

4.3 Results on M3 and LncRNA data In this experiment, we make label predictions to explore the performance of DeepM4L on M3 data from the bag



(a) Performance vs. C (b) Loss vs. Epoch

Figure 3: (a) The impact of the number of different layers C on AvgF1 and AUPRC values for the Isoform dataset. (b) The convergence trend of the objective function on the LncRNA and MSRC v2 datasets.

level and the instance level, respectively. For the four canonical single-view multi-instance and multi-label datasets listed in Table 2, we randomly divide the data view into two views, each of which contains half of the features of the original view. The other preprocessing is the same as in the previous subsection. We report the experimental results at the bag-level and the instance-level in Tables 4 and 5, respectively.

DeepM4L always has the relatively high AvgF1 and AUPRC values among the compared methods, and DeepM4L improves AvgF1 by at least 1% w.r.t. the second best baseline in mean. From the comprehensive view of the performance of DeepM4L is better. The AvgF1 value improvement is more significant with respect to shallow matrix factorization based methods. This observation indicates deep heterogeneous network embedding can effectively capture network structure and attribute information. These results again confirm the effectiveness of DeepM4L on predicting the labels of bags and of individual instances.

We further study the performance of DeepM4L on the LncRNA dataset (a natural testbed for heterogeneous network embedding) [14], and report the results in Table 6. DeepM4L always has the highest AvgF1 and AUPRC values among the compared methods. This experiment further proves the effectiveness of DeepM4L on modeling complex biological network data.

4.4 Parameter Sensitivity Analysis In this paper, the parameters (number of layers C) should be specified in advance. We report the changes of AvgF1 and AUPRC values under different C in Fig. 3(a). It can be seen that as the number of layers increases, their values gradually increase. This observation suggests that instance embedding should be leveraged to mine the network structure and attribution information. However, when $C \geq 7$, over-fitting occurs and causes a performance degradation. We also record the objective function output value in Fig 3(b). We can see that the loss decreases as the iteration proceeds and comes to a convergence within 10 iterations. This trend proves that our method can quickly converge.

5 Conclusions

In this paper, we studied a novel learning method (Deep Multi-typed objects Multi-view Multi-instance Multi-label Learning) for naturally interconnected multi-typed complex objects. Experimental results on real-world and benchmark datasets validated that DeepM4L can more comprehensively fuse multi-typed objects and mine complex relations between bags, instances and labels, and it achieves better results than other competitive and related methods. Our work presents a showcase of mining complex objects by unifying heterogeneous network embedding and multi-instance learning.

References

[1] Cam-Tu Nguyen, De-Chuan Zhan, and Zhi-Hua Zhou. Multi-modal image annotation with multi-instance multi-label lda. In *IJCAI*, pages 1558–1564, 2013.

[2] Yang Yang, Yi-Feng Wu, De-Chuan Zhan, Zhi-Bin Liu, and Yuan Jiang. Complex object classification: A multi-modal multi-instance multi-label deep network with optimal transport. In *KDD*, pages 2594–2603, 2018.

[3] Yuying Xing, Guoxian Yu, Carlotta Domeniconi, Jun Wang, Zili Zhang, and Maozu Guo. Multi-view multi-instance multi-label learning based on collaborative matrix factorization. In *AAAI*, pages 5508–5515, 2019.

[4] Jing Zhao, Xijiong Xie, Xin Xu, and Shiliang Sun. Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38:43–54, 2017.

[5] Guoxian Yu, Hailong Zhu, Carlotta Domeniconi, and Maozu Guo. Integrating multiple networks for protein function prediction. In *BMC Systems Biology*, volume 9, page S3, 2015.

[6] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *JMLR*, 12:2211–2268, 2011.

[7] Vladimir Gligorijević and Nataša Pržulj. Methods for biological data integration: perspectives and challenges. *Journal of the Royal Society Interface*, 12(112):20150571, 2015.

[8] Marinka Žitnik and Blaž Zupan. Data fusion by matrix factorization. *TPAMI*, 37(1):41–53, 2015.

[9] Guangyuan Fu, Jun Wang, Carlotta Domeniconi, and Guoxian Yu. Matrix factorization-based data fusion for the prediction of lncrna–disease associations. *Bioinformatics*, 34(9):1529–1537, 2018.

[10] Yuehui Wang, Guoxian Yu, Carlotta Domeniconi, Jun Wang, Xiangliang Zhang, and Maozu Guo. Selective matrix factorization for multi-relational data fusion. In *DASFAA*, pages 313–329, 2019.

[11] Keyao Wang, Jun Wang, Carlotta Domeniconi, Xiangliang Zhang, and Guoxian Yu. Differentiating isoform functions with collaborative matrix factorization. *Bioinformatics*, 36(6):1864–1871, 2020.

[12] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. Representation learning for attributed multiplex heterogeneous network. In *KDD*, pages 1358–1368, 2019.

[13] Binbin Hu, Yuan Fang, and Chuan Shi. Adversarial learning on heterogeneous information networks. In *KDD*, page 120–129, 2019.

[14] Guoxian Yu, Yuehui Wang, Jun Wang, Carlotta Domeniconi, Maozu Guo, and Xiangliang Zhang. Attributed heterogeneous network fusion via collaborative matrix tri-factorization. *Information Fusion*, 63:153–165, 2020.

[15] Yuanlin Yang, Guoxian Yu, Jun Wang, Carlotta Domeniconi, and Xiangliang Zhang. Multi-type objects multi-view multi-instance multi-label learning. *ICDM*, pages 1–6, 2020.

[16] Xia Chen, Guoxian Yu, Jun Wang, Carlotta Domeniconi, Zhao Li, and Xiangliang Zhang. Activehne: active heterogeneous network embedding. In *IJCAI*, pages 2123–2129, 2019.

[17] Yizhou Zhang, Yun Xiong, Xiangnan Kong, Shanshan Li, Jinhong Mi, and Yangyong Zhu. Deep collective classification in heterogeneous information networks. In *WWW*, pages 399–408, 2018.

[18] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. Heterogeneous network representation learning: Survey, benchmark, evaluation, and beyond. *TKDE*, 99(1):1–19.

[19] Yuying Xing, Guoxian Yu, Jun Wang, Carlotta Domeniconi, and Xiangliang Zhang. Weakly-supervised multi-view multi-instance multi-label learning. In *IJCAI*, pages 3124–3130, 2020.

[20] Bing Li, Chunfeng Yuan, Weihua Xiong, Weiming Hu, Houwen Peng, Xinmiao Ding, and Steve Maybank. Multi-view multi-instance learning based on joint sparse representation and multi-view dictionary learning. *TPAMI*, 39(12):2554–2560, 2017.

[21] Qiaoyu Tan, Guoxian Yu, Jun Wang, Carlotta Domeniconi, and Xiangliang Zhang. Individuality- and commonality-based multiview multilabel learning. *IEEE TCYB*, 99(1):1–12, 2021.

[22] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*, pages 135–144, 2017.

[23] Bingbing Xu, Huawei Shen, Qi Cao, Keting Cen, and Xueqi Cheng. Graph convolutional networks using heat kernel for semi-supervised learning. In *IJCAI*, pages 1928–1934, 2019.

[24] Qiaoyu Tan, Guoxian Yu, Carlotta Domeniconi, Jun Wang, and Zili Zhang. Incomplete multi-view weak-label learning. In *AAAI*, pages 4414–4421, 2018.

[25] Zhi-Hua Zhou, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. Multi-instance multi-label learning. *Artificial Intelligence*, 176(1):2291–2320, 2012.

[26] Gerson Zaverucha and Santos Costa Vítor. Guest editors' introduction: special issue on inductive logic programming and on multi-relational learning. *Machine Learning*, 100:1–3, 2015.

[27] Cam-Tu Nguyen, Xiaoliang Wang, Jing Liu, and Zhi-Hua Zhou. Labeling complicated objects: Multi-view multi-instance multi-label learning. In *AAAI*, 2014.

[28] Bin Wu, Erheng Zhong, Andrew Horner, and Qiang Yang. Music emotion recognition by multi-label multi-layer multi-instance multi-view learning. In *ACM MM*, page 117–126, 2014.

[29] Yang Yang, Zhao-Yang Fu, De-Chuan Zhan, Zhi-Bin Liu, and Yuan Jiang. Semi-supervised multi-modal multi-instance multi-label deep network with optimal transport. *TKDE*, 99(1):1–14, 2021.

[30] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *TBD*, 6(1):3–28, 2020.

[31] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, pages 701–710, 2014.

[32] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.

[33] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C. Aggarwal, and Thomas S. Huang. Heterogeneous network embedding via deep architectures. In *KDD*, page 119–128, 2015.

[34] Ji Feng and Zhi-Hua Zhou. Deep miml network. In *AAAI*, pages 1884–1890, 2017.

[35] Forrest Briggs, Xiaoli Z Fern, and Raviv Raich. Rank-loss support instance machines for miml instance annotation. In *KDD*, pages 534–542, 2012.

[36] Guoxian Yu, Xia Chen, Carlotta Domeniconi, Jun Wang, Zhao Li, Zili Zhang, and Xiangliang Zhang. Cmal: Cost-effective multi-label active learning by querying subexamples. *TKDE*, 99(1):1–14, 2021.

[37] Sheng-Jun Huang, Nengneng Gao, and Songcan Chen. Multi-instance multi-label active learning. In *IJCAI*, pages 1886–1892, 2017.

[38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[39] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.

[40] Qiuyue Huang, Jun Wang, Xiangliang Zhang, and Guoxian Yu. Isoform-disease association prediction by data fusion. In *ISBRA*, pages 44–55, 2020.

[41] Guoxian Yu, Keyao Wang, Carlotta Domeniconi, Maozu Guo, and Jun Wang. Isoform function prediction based on bi-random walks on a heterogeneous network. *Bioinformatics*, 36(1):303–310, 2020.

[42] Minling Zhang and Zhihua Zhou. A review on multi-label learning algorithms. *TKDE*, 26(8):1819–1837, 2014.