Learning with Augmented Multi-Instance View

Yue Zhu
Jianxin Wu
Yuan Jiang
Zhi-Hua Zhou

National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023
Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023

Abstract

In this paper, we propose the Augmented Multi-Instance View (AMIV) framework to construct a better model by exploiting augmented information. For example, abstract screening tasks may be difficult because only abstract information is available, whereas the performance can be improved when the abstracts of references listed in the document can be exploited as augmented information. If each abstract is represented as an instance (i.e., a feature vector) $x$, then with the augmented information, it can be represented as an instance-bag pair $(x, B)$, where $B$ is a bag of instances (i.e., the abstracts of references). Note that if $x$ has a label $y$, then we assume that there must exist at least one instance in the bag $B$ having the label $y$. We regard $x$ and $B$ as two views, i.e., a single-instance view augmented with a multi-instance view, and propose the AMIV-lss approach by establishing a latent semantic subspace between the two views. The AMIV framework can be applied when the augmented information is presented as multi-instance bags and to the best of our knowledge, such a learning with augmented multi-instance view problem has not been touched before. Experimental results on twelve TechPaper datasets, five PubMed data sets and a WebPage data set validate the effectiveness of our AMIV-lss approach.

Keywords: learning with Augmented multi-instance view, multi-view learning, multi-instance learning, semantic subspace learning, non-negative matrix factorization

1. Introduction

Abstract screening is widely used in academic paper searching applications, whose target is to tell whether a paper is related to a given topic with only abstract information available (C.-E. Brodley et al., 2012; Cohen et al., 2006). Such a binary classification task is difficult sometimes, because the words characterizing the document label might not appear in the abstract, due to the fact that authors want to stress more about their contributions in limited space. Fortunately, many online publishers provide service in the style of “Abstract + Reference free, Fulltext pay”, and thus the abstracts of references can be obtained as augmented information to improve the classification performance.

Many real applications have shown that learning performance can be improved through augmented information. For example, in an image annotation task, tags surrounding an image can be used as augmented information to facilitate the annotation. In a recommender system, both user relationship and item relationship can be regarded as augmented information to improve the recommendation performance. Some well-studied learning paradigms, including multi-modal learning and transfer learning, have demonstrated the effectiveness taking advantage of augmented information. In multi-modal learning paradigm, such augmented information is presented as features.
derived from different modalities, while in transfer learning it is presented as the source domain from where knowledge are transferred to the target domain.

However, those approaches taking advantages of augmented information cannot be directly applied in this case. A specific property of abstract screening tasks with augmented information lies in the fact that, if we regard the original data as an instance (e.g., by representing the abstract as a TF-IDF feature vector), then the augmented information is a bag of other instances (i.e., a set of TF-IDF feature vectors with each corresponding to the abstract of one reference). Moreover, it is clearly not the case that all instances in the augmented bag belong to the same category as the original instance. A more reasonable assumption is that in this augmented bag, there is at least one instance belonging to the same category as the original instance. Besides, for different instances, the number of instances in their augmented bags can be different, which corresponds to the fact that different documents usually have different number of references.

We regard the original data in a single-instance view and the augmented data in a multi-instance view, thus propose the Augmented Multi-Instance View (AMIV) learning framework for tasks where augmented information can be used, and such a learning with augmented multi-instance view problem has not been studied before. Under the AMIV framework, we propose the $AMIV_{lss}$ approach to deal with classification tasks by establishing a latent semantic subspace (LSS) in which the representation of a single-instance view instance and the representation of its corresponding augmented multi-instance view bag are close to each other.

Such AMIV framework can be applied not only in abstract screening tasks, but other situations where the augmented information is presented as multi-instance bags. For example, it is difficult to classify some twitter texts as they are too short, whereas their follow-up twitter discussions can be used as helpful augmented multi-instance view. Another example is web page classification, where we can regard the linked web pages as augmented multi-instance view for classifying a web page. Experimental results on two types of text categorization tasks, i.e., abstract screening tasks (including 12 TechPaper and 5 PubMed data sets) and web classification tasks (WebPage data set), validate the effectiveness of our approach making use of augmented multi-instance view.

The rest of this paper is organized as following. In Section 2, a brief review of related work is given. Then, the proposed learning framework AMIV and AMIV-lss approach are presented in Section 3. In Section 4, experimental results of the $AMIV_{lss}$ approach together with several compared approaches are reported on abstract screening tasks and web classification tasks. Finally, we conclude our work and raise issues for future work in Section 5.

2. Related Work

In this section, we first introduce some state-of-the-art approaches successfully taking advantage of augmented information, then make a brief survey on related work of multi-instance learning, multi-view learning and multi-instance multi-view learning.

By leveraging augmented information in learning, one can expect a better performance to be achieved. Such an idea has been fulfilled by several learning paradigms from different perspectives, including multi-modal learning, transfer learning, etc. In multi-modal learning, different modalities can be regarded as augmented sources of information to each other. For example, in image annotation, images and surrounding text are two different modalities (Nguyen et al., 2013), and by using them together, a better performance will be achieved. In transfer learning, data in a different but relevant source domain are utilized as augmented information to transfer knowledge to target do-
main (Pan and Yang, 2010). Indoor WiFi localization for instance, the signal data collected in one
time period may differ from those collected in another, thus a transfer learning method is necessary
in order to exploit information in a different domain (Pan et al., 2011). In applications on recom-
mender system, Pan et al. (2010) proposed a transfer learning method to improve recommendation
performance leveraging augmented information about both users and items. Besides, many studies
try to incorporate data structure, which can also be regarded as a term of augmented information.
Laplacian SVM, proposed by Belkin et al. (2006), is a successful example making use of manifold
structure in both labeled and unlabeled data. However, the above approaches cannot be directly
applied in the AMIV framework, which contains a single-instance and a multi-instance view.

Multi-instance learning (MIL) was firstly proposed by Dietterich et al. (1997) in their study
on drug activity prediction. After that, many literatures are devoted to MIL, and many algorithms
have been developed. To name a few, Maron and Lozano-Pérez (1997) presented the DD algorithm
to identify the concept point and classify bags by distance between instances in the bag and the
concept point. Andrews et al. (2002) generalized support vector machine algorithm to MIL via
imposing additional MIL constraints. Chen et al. (2006) proposed MILES, transforming the multi-
instance learning into single-instance learning via embedding each bag into feature space defined by
instances in all training bags. Zhou et al. (2009) treated instances in bags as non-i.i.d samples and
proposed miGraph method to take advantage of instance relationship in the bag. Foulds and Frank

Multi-view learning deals with data described in multiple views, i.e., multiple feature sets. Here,
each view is a feature set, whereas for each example there is a corresponding instance in each view.
The goal is to exploit the relation between the multiple views to improve performance or reduce
the sample complexity. Multi-view learning has been well studied in learning with unlabeled da-
ta. Indeed the most famous multi-view learning algorithm, co-training (Blum and Mitchell, 1998),
is a semi-supervised learning algorithm. Although it has been shown that disagreement-based ap-
proaches, such as co-training, do not really need the existence of multiple views, given that there
are learners with sufficient diversity (Wang and Zhou, 2007, 2010b). It has also been shown that
with suitable multiple views, semi-supervised learning with a single labeled example can be pos-
sible (Zhou et al., 2007). Thus, the multiple views really help. It has also been proved that by
exploiting multiple views, active learning is able to achieve exponential sample complexity im-
provement under non-realizable case (Wang and Zhou, 2010a). Multi-view learning also provides
a natural way to combine active learning with semi-supervised learning (Wang and Zhou, 2008).
Note that many studies in multi-view learning tried to establish a latent subspace by assuming that
instances (in different views) belong to the same example are nearby after mapping into the same
latent subspace.

There are several pieces of work combining multi-instance learning and multi-view learning
together. Mayo and Frank (2011) empirically investigated the benefits of multi-view multi-instance
learning for supervised image classification, which trains multi-instance classifiers in each view and
then ensembles them together to build the final classifier. Li et al. (2012) proposed a general method,
which in every iteration, trains a classifier with a label candidate set in each view by multiple ker-
nel learning and then updates the label candidate set via labels predicted by classifiers trained in the
other views. Zhang et al. (2013) proposed MIT2LS to solve multi-instance learning from multiple in-
formation sources, which minimizes the structure risk plus the multi-instance classification loss and
the penalty on the inconsistency of the classifier in different views of the same example. Nguyen
et al. (2013) proposed M3LDA, a multi-view multi-instance multi-label (MIML) approach, which
is able to exploit multi-modal information in MIML. All the aforementioned approaches work on multiple views, and in each an example is represented as a bag of instances or in other words, they work with two multi-instance views. In contrast, though there are two views in our AMIV setting, there is only one multi-instance view, i.e. the “reference” part. If one regards the “abstract” part as a multi-instance view, then the corresponding “reference” part will be a “set of multi-instance bags” (each bag represents an abstract of a reference) and thus, leading to an one-layer multi-instance view (the “abstract” part) and a two-layer multi-instance view (the “reference” part). This will be too complicated than necessary.

In this paper, we propose to exploit augmented information to help the original supervised learning, where the augmented information is a multi-instance view. In contrast to previous studies on multi-view multi-instance learning, in our task we have one single-instance view (the original information) and one multi-instance view (the augmented information). To the best of our knowledge, there is no study on such heterogeneous multi-view multi-instance learning before.

3. The AMIV Framework

In this section we propose the AMIV framework to solve learning with the Augmented Multi-Instance View problem via establishing a latent semantic subspace (LSS) where the representation of a single-instance view instance and the representation of its corresponding augmented multi-instance view bag are close to each other.

3.1. The Formulation

Before introducing the approach, we formulate the AMIV (learning with Augmented Multi-Instance View) framework as follows. Let $\mathcal{X}_S$ denote the instance space (or feature space) of the original single-instance view, $\mathcal{X}_A$ denote the instance space (or feature space) of the augmented multi-instance view and $\mathcal{Y}$ denote the set of class labels. The task is to learn a function $f : (\mathcal{X}_S; 2^{\mathcal{X}_A}) \rightarrow \mathcal{Y}$ from a given data set $D = \{(x_i, B_i, y_i) | i = 1, 2, \cdots, n\}$, which maps a single instance to a class label with an augmented multi-instance view, to predict an unseen instance $x_{\text{new}}$’s label. Here $x_i \in \mathcal{X}_S$ is an instance in the original single-instance view, $B_i = \{b_{i1}, b_{i2}, \cdots, b_{in_i}\} \subseteq \mathcal{X}_A$ is the corresponding bag of size $n_i$ in the augmented multi-instance view and $y_i \in \mathcal{Y}$ is the corresponding class label.

3.2. The AMIV-lss Approach

A common subspace which captures the intrinsic structure of data across multiple views will greatly alleviate the difficulty of the learning task (Guo, 2013). Inspired by such an idea, we propose the AMIV-lss approach under the AMIV framework, which handles classification in a latent semantic subspace (LSS) denoted as $\mathcal{J}$ where the representation of a single-instance view instance $x$ and the representation of its corresponding augmented multi-instance view bag $B$ is close to each other. Directly specify representation for $B$ is difficult because there are multiple instances with unknown labels in $B$, therefore we consider to find a prototype $s$ for $B$ which can decide the bag label, and then the bag representation is transformed to the representation of $s$ in LSS $\mathcal{J}$. A two stage optimization strategy is adopted: an optimal latent semantic subspace is learned in the first stage, and then a maximum margin classifier is trained in the second stage.
3.2.1. LEARNING INSTANCE REPRESENTATION

Let \( X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d \times n} \) denote \( n \) instances in the single-instance view, and let \( P_X = [p_{x_1}, p_{x_2}, \ldots, p_{x_n}] \in \mathbb{R}^{t \times n} \) denote the representation of \( X \) in LSS \( J \) where \( p_{x_i} \) is the representation of \( x_i \). Denote by \( Q = [q_1, q_2, \ldots, q_l]^\top \in \mathbb{R}^{t \times d} \) the basis of LSS \( J \), then we have \( X^\top = P_X^\top Q \).

Such a representation is useful for real applications, especially those with text (Foltz et al., 1998). Specifically, \( X \) is an item-word matrix which indicates the semantic relationship between items and words, and \( X^\top = P_X^\top Q \) models such semantic relationship by interaction between semantic topic patterns of items and word patterns of semantic topics, where \( p_{x_i} \) in \( P_X \) stands for the pattern of \( t \) semantic topics of \( x \) and \( q \) in \( Q \) represents the word present pattern of a semantic topic.

Because \( P_X \) and \( Q \) represent semantic topic patterns and word present patterns respectively, both of them should be non-negative. Note that only small quantities of semantic topics are related to main topic of an item and words presented in a certain semantic topic are much fewer than those in the whole dictionary, thus both \( P_X \) and \( Q \) should be sparse.

In order to establish the latent semantic subspace (LSS) and obtain representations of instances in that space, we follow a non-negative matrix factorization (NMF) framework (Hsieh and Dhillon, 2011) as

\[
\min_{P_X \geq 0, Q \geq 0} \left\| X^\top - P_X^\top Q \right\|_F^2 + \Omega(P_X, Q) \tag{1}
\]

where \( \Omega(P_X, Q) = \left\| P_X \right\|_1 + \left\| Q \right\|_1 \).

Let \( Z = [B_1, B_2, \ldots, B_n] \in \mathbb{R}^{d \times r} \) denote the instances in the augmented multi-instance view where \( r = \sum_{i=1}^n n_i \) and \( P_Z = [P_{B_1}, P_{B_2}, \ldots, P_{B_n}] \in \mathbb{R}^{t \times r} \) denote the representation of \( Z \) in the LSS \( J \) where \( P_{B_i} = [p_{b_{i1}}, p_{b_{i2}}, \ldots, p_{b_{ini}}] \) with each \( p_{b_{ij}} \) denoting the representation of \( b_{ij} \). Similarly, we have

\[
\min_{P_Z \geq 0, Q \geq 0} \left\| Z^\top - P_Z^\top Q \right\|_F^2 + \Omega(P_Z, Q). \tag{2}
\]

3.2.2. LEARNING BAG REPRESENTATION

Although instance representation in the LSS \( J \) is given, one still cannot specify the representation of a bag directly in that space, because there are multiple instances with unknown labels in that bag. To identify the bag representation, we first define a key prototype \( s \) for bag \( B \) which can decide the \( B \)'s label. Let \( S = [s_1, s_2, \ldots, s_n] \in \mathbb{R}^{d \times n} \) denote the space where the key prototypes live, with each \( s_i \) representing the key prototype of \( B_i \) and let \( P_S = [p_{s_1}, p_{s_2}, \ldots, p_{sn}] \in \mathbb{R}^{t \times n} \) denote the representation of \( S \). That is, \( p_{si} \) is the bag representation of \( B_i \). Similarly to Eq.(2), we have

\[
\min_{P_S \geq 0, Q \geq 0} \left\| S^\top - P_S^\top Q \right\|_F^2 + \Omega(P_S, Q). \tag{3}
\]

One straightforward idea for specifying the key prototype of a bag is to take the center of instances in that bag. However, this strategy may be risky, e.g., if a positive bag hold much more negative instances than positive ones, then the center of it will be negative, just the opposite to the bag label. In this case, taking center will be misleading and may cause degeneration of performance. To reduce this risk, instance specific weighting can be taken into account. Given \( x \) in the single-instance view, the closer an instance in corresponding bag \( B \) is to \( x \), the higher probability it will have the same label as \( x \), thus the larger instance weight it should take. There comes another
problem that distance comparison among instances becomes unreliable, when met the situation that instances are of high dimension and sparsity.

In order to handle the above problems, a locally linear assumption (Roweis and Saul, 2000) is brought in to specify key prototypes. We assume that prototype \( s_i \) of bag \( B_i \) is the linear combination of intra bag instances in the neighborhood of \( x_i \), where the neighborhood is defined in the LSS \( J \). We define neighbor indicator vector of \( B_i \) as \( \delta_i \) and \( k \) nearest neighbors as \( N_k \). Thus, if \( p_{b_{ij}} \in N_k(x_i) \), then \( \delta_{ij} = 1 \); otherwise, \( \delta_{ij} = 0 \). Let \( \alpha_i \) be the linear combination coefficient vector of \( B_i \), then we have \( s_i = B_i \alpha_i \). Each coefficient in \( \alpha_i \) is decided by distance inverse weighting, \( \alpha_{ij} = \exp(-\text{dist}_{ij}^2) \delta_{ij}/(\sum_{j=1}^n \exp(-\text{dist}_{ij}^2) \delta_{ij}) \), where \( \exp \) is the exponential function and \( \text{dist}_{ij} \) is the Euclidean distance between \( p_{xi} \) and \( p_{b_{ij}} \), \( dist_{ij} = \| p_{xi} - p_{b_{ij}} \|_2 \). We rewrite \( S \) as

\[
S = [s_1, s_2, \ldots, s_n], \text{ with } s_i = \sum_{j=1}^n \frac{\exp(-\| p_{xi} - p_{b_{ij}} \|_2^2) \delta_{ij}}{\sum_{j=1}^n \exp(-\| p_{xi} - p_{b_{ij}} \|_2^2) \delta_{ij}}, \forall s_i \in S. \tag{4}
\]

### 3.2.3. Two Stage Optimization

We will propose a two stage optimization strategy. In the first stage, we learn an optimal latent semantic subspace where representation of a single instance view instance \( x \) and that of its corresponding multi-instance view bag \( B \) is similar. In the second stage, we train a classifier in that space. Defining \( P = [X, P_Z, P_S] \) and combining Eqs.(1)-(3) together, we have

\[
\min_{P \geq 0, Q \geq 0} \left\| [X, Z, S]^\top - [P_X, P_Z, P_S]^\top Q \right\|_F^2 + \Omega(P, Q). \tag{5}
\]

As previously mentioned, in the LSS \( J \), the representation of single-instance view instance and that of its corresponding augmented multi-instance view bag should be close to each other. Namely, \( P_X \) and \( P_S \) should be similar, and thus we take a \( \| P_X - P_S \|_F^2 \) term besides \( \| P \|_1 \) and \( \| Q \|_1 \) in \( \Omega(P, Q) \). Then, the first stage optimization casts as

\[
(P^*, Q^*) = \arg \min_{P \geq 0, Q \geq 0} \left\| [X, Z, S]^\top - P^\top Q \right\|_F^2 + \lambda_1 \| P_X - P_S \|_F^2 + \lambda_2 \left( \| P \|_1 + \| Q \|_1 \right), \tag{6}
\]

where \( Q^* \) is the basis of the optimal latent semantic subspace \( J^* \), and \( P^* = [P^*_X, P^*_Z, P^*_S] \) is the representations in \( J^* \), and \( \lambda_1 \) and \( \lambda_2 \) are two parameters to trade off.

In the second stage, in order to handle classification task in LSS \( J^* \), we train a maximum margin classification model \( w^* \) with an additional constraint that representation of a prototype \( s_i \) should hold the same label as representation of corresponding single-instance view instance \( x_i \). The second stage optimization casts as

\[
w^* = \arg \min_w \frac{1}{2} \| w \|_2^2 + C \sum_{i=1}^n \xi_i \quad \text{s.t.} \quad \begin{align*}
y_i w^\top p_{xi}^* & \geq 1 - \xi_i \\
y_i w^\top p_{s_i}^* & \geq 1 - \xi_i \\
\xi_i & \geq 0, \quad i = 1, 2, \ldots, n
\end{align*} \tag{7}
\]
Specifically, in each iteration, to learn the bag representation we firstly update prototypes \( S \) optimize with respect to all of them jointly. Therefore, an alternative optimization is applied to solve Eq.(6). Output:

\[ Q, \]

Process:

1. Initialize \( P_X, P_Z \) and \( Q \) ← Eq.(10);
2. repeat:
   3. Update prototypes \( S \) ← Eq.(4);
   4. Fix \( S \) and \( P_Z \), update \( P_X, P_S \) and \( Q \) ← Eq.(8);
   5. Fix \( S, P_X, P_S \) and \( Q \), update \( P_Z \) ← Eq.(9);
3. until the maximum iteration is reached.
7. Given \( P_X \) and \( P_S \), solve \( w \) ← Eq.(7);

### Algorithm 1 The AMIV-lss Approach

**Input:** Training Data \((x_i, B_i, y_i), i = 1, 2, \cdots , n\)

**Output:** \( Q, w \)

**Process:**

1. Initialize \( P_X, P_Z \) and \( Q \) ← Eq.(10);
2. repeat:
   3. Update prototypes \( S \) ← Eq.(4);
   4. Fix \( S \) and \( P_Z \), update \( P_X, P_S \) and \( Q \) ← Eq.(8);
   5. Fix \( S, P_X, P_S \) and \( Q \), update \( P_Z \) ← Eq.(9);
6. until the maximum iteration is reached.
7. Given \( P_X \) and \( P_S \), solve \( w \) ← Eq.(7);

#### 3.2.4. Optimization Algorithm

In the first stage, the objective function in Eq.(6) involves \( S, P_X, P_Z, P_S \) and \( Q \), and it is not easy to optimize with respect to all of them jointly. Therefore, an alternative optimization is applied to solve Eq.(6). Specifically, in each iteration, to learn the bag representation we firstly update prototypes \( S \) via Eq.(4). For fixed \( (X, S) \)\):

\[
\min_{P_X \geq 0, P_Z \geq 0, Q \geq 0} \left[ [X, S] - [P_X, P_Z]^{\top} Q \right]_F^2 + \lambda_1 \left[ P_X - P_S \right]_F^2 + \lambda_2 \left[ P_X \right]_1 + \left[ P_S \right]_1 + \left[ Q \right]_1, \tag{8}
\]

and we update \( P_X, P_S \) and \( Q \) by solving Eq.(8). Finally, with fixed \( S, P_X, P_S \) and \( Q \), optimizing Eq.(6) is equivalent to learning the representation of \( Z \) as:

\[
\min_{P_Z \geq 0} \left[ Z - P_Z^{\top} Q \right]_F^2 + \lambda_2 \left[ P_Z \right]_1, \tag{9}
\]

therefore we update \( P_Z \) via solving Eq.(9). After the first stage is finished, with optimal \( P_X^* \) and \( P_S^* \) obtained, we learn the optimal classifier \( w^* \) according to Eq.(7) in the second stage.

Moreover, in order to get the algorithm started, we initialize \( P_X, P_Z \) and \( Q \) via solving

\[
\min_{P_X \geq 0, P_Z \geq 0, Q \geq 0} \left[ [X, Z] - [P_X, P_Z]^{\top} Q \right]_F^2 + \lambda_2 \left[ P_X \right]_1 + \left[ P_Z \right]_1 + \left[ Q \right]_1, \tag{10}
\]

and Algorithm 1 summarizes the AMIV-lss approach.

In further detail, we iteratively solve one variable with the other two variables fixed, so as to solve Eq.(8) whose objective function is hard to optimize with respect to \( P_X, P_S \) and \( Q \) jointly. With \( P_X \) and \( Q \) fixed, we solve Eq.(8) with respect to \( P_S \) by solving Eq.(11) equivalently.

\[
\min_{P_S \geq 0} \left[ S^{\top} - P_S^{\top} Q \right]_F^2 + \lambda_1 \left[ P_S - P_X \right]_F^2 + \lambda_2 \left[ P_S \right]_1. \tag{11}
\]

Then, with \( P_S \) and \( Q \) fixed, we solve Eq.(8) with respect to \( P_X \) via solving Eq.(12) equivalently.

\[
\min_{P_X \geq 0} \left[ X^{\top} - P_X^{\top} Q \right]_F^2 + \lambda_1 \left[ P_X - P_S \right]_F^2 + \lambda_2 \left[ P_X \right]_1. \tag{12}
\]

Finally, with \( P_X \) and \( P_S \) fixed, we solve Eq.(8) with respect to \( Q \) by equivalently solving

\[
\min_{Q \geq 0} \left[ [X, S]^{\top} - [P_X, P_S]^{\top} Q \right]_F^2 + \lambda_2 \left[ Q \right]_1. \tag{13}
\]
In order to solve Eq.(10), similarly to Eq.(8), we iteratively fix \([P_X, P_Z]\), solve Eq.(10) with respect to \(Q\), and fix \(Q\), solve Eq.(10) with respect to \([P_X, P_Z]\). More specifically, the former is equivalent to solving \(Q\) by
\[
\min_{Q \geq 0} \left\| \begin{bmatrix} X, Z \end{bmatrix}^\top - [P_X, P_Z]^\top Q \right\|^2_F + \lambda_2 \| Q \|_1, \tag{14}
\]
and the latter is equivalent to solving \([P_X, P_Z]\) via
\[
\min_{[P_X, P_Z] \geq 0} \left\| \begin{bmatrix} X, Z \end{bmatrix}^\top - [P_X, P_Z]^\top Q \right\|^2_F + \lambda_2 \| [P_X, P_Z] \|_1. \tag{15}
\]

In order to deal with optimization for Eqs.(8)-(15), a greedy coordinate descend method is applied (Hsieh and Dhillon, 2011), that one variable update rule is as
\[
P_{ir} \leftarrow \max(0, P_{ir} - G_{P_{ir}} / H_{P_{ii}}),
Q_{ir} \leftarrow \max(0, Q_{ir} - G_{Q_{ir}} / H_{Q_{ii}}), \tag{16}
\]
where \(G\) is the gradient of the objective function \(f(P, Q)\) and \(H\) is the Hessian of \(f(P, Q)\).

In order to solve Eq.(7), firstly we construct a new set of data organized as \([P_X, P_S]\) and corresponding label vector as \(\begin{bmatrix} y \ y \end{bmatrix}\). Then liblinear (Fan et al., 2008) is applied to get the optimal solution \(w^*\).

In prediction stage, given a new instance \(x_{new}\) and previously learned \(Q^*\) and \(w^*\), our task is to predict the corresponding label \(y_{new}\). Firstly, we obtain the representation of \(x_{new}\) in \(J^*\) via solving
\[
P_{x_{new}}^* = \arg \min_{P_{x_{new}} \geq 0} \left\| x_{new}^\top - P_{x_{new}}^\top Q^* \right\|^2_F + \lambda_2 \| P_{x_{new}} \|_1.
\]
Then we predict \(y_{new}\) by
\[
y_{new} = \text{sign}(w^\top P_{x_{new}}^*).
\]

4. Experiment

In this section, we evaluate our approach AMIV-lss on abstract screening tasks (i.e. the TechPaper and the Pubmed data sets) and web page classification tasks (i.e. the WebPage data set). The experimental results validate the effectiveness of the AMIV-lss approach which takes advantage of augmented multi-instance view.

4.1. Data Sets

We construct a number of applications of abstract screening tasks, including the TechPaper data sets and PubMed data sets. Moreover, we also build the WebPage data sets from web page classification applications, so as to show that our framework and approach can be applied to other application fields with augmented information presented as multi-instance bags. The data sets will be publicly available.
Table 1: TechPaper Data Sets. #obj denotes the number of instances in the single-instance view (also the number of bags in the augmented multi-instance view); #inst represents the total number of instances in both single-instance and augmented multi-instance views; #inst/bag is the average number of instances per bag in the augmented multi-instance view.

<table>
<thead>
<tr>
<th></th>
<th>AL</th>
<th>CLST</th>
<th>DPL</th>
<th>MTCL</th>
<th>MIL</th>
<th>MLL</th>
<th>MTL</th>
<th>MVL</th>
<th>OL</th>
<th>RL</th>
<th>SSL</th>
<th>TL</th>
</tr>
</thead>
<tbody>
<tr>
<td>#obj</td>
<td>462</td>
<td>378</td>
<td>354</td>
<td>418</td>
<td>398</td>
<td>418</td>
<td>458</td>
<td>438</td>
<td>376</td>
<td>448</td>
<td>400</td>
<td></td>
</tr>
<tr>
<td>#inst</td>
<td>8,812</td>
<td>7,468</td>
<td>6,048</td>
<td>8,280</td>
<td>7,470</td>
<td>7,971</td>
<td>8,69</td>
<td>8,238</td>
<td>9,803</td>
<td>6,983</td>
<td>8,184</td>
<td>7,361</td>
</tr>
<tr>
<td>#inst/bag</td>
<td>18.07</td>
<td>18.76</td>
<td>16.08</td>
<td>18.81</td>
<td>17.81</td>
<td>17.77</td>
<td>18.07</td>
<td>17.05</td>
<td>16.82</td>
<td>17.57</td>
<td>17.27</td>
<td>17.40</td>
</tr>
</tbody>
</table>

Table 2: PubMed Data Sets. #obj denotes the number of instances in the single-instance view (also the number of bags in the augmented multi-instance view); #inst represents the total number of instances in both single-instance and augmented multi-instance views; #inst/bag is the average number of instances per bag in the augmented multi-instance view.

<table>
<thead>
<tr>
<th></th>
<th>ADHD</th>
<th>Antihistamines</th>
<th>Estrogens</th>
<th>Oral Hypoglycemics</th>
<th>Urinary Incontinence</th>
</tr>
</thead>
<tbody>
<tr>
<td>#obj</td>
<td>849</td>
<td>306</td>
<td>368</td>
<td>422</td>
<td>327</td>
</tr>
<tr>
<td>#inst</td>
<td>16,674</td>
<td>6,021</td>
<td>7,213</td>
<td>8,222</td>
<td>6,462</td>
</tr>
<tr>
<td>#inst/bag</td>
<td>19.64</td>
<td>19.67</td>
<td>19.60</td>
<td>19.48</td>
<td>19.76</td>
</tr>
</tbody>
</table>

4.1.1. The 12 TechPaper Data Sets

We select 12 topics in machine learning fields, including active learning (AL), clustering (CLST), deep learning (DPL), metric learning (MTCL), multi-instance learning (MIL), multi-label learning (MLL), multi-task learning (MTL), multi-view learning (MVL), online learning (OL), reinforcement learning (RL), semi-supervised learning (SSL) and transfer learning (TFL). Then we download abstracts of both academic papers on those topics (for single-instance view) and corresponding references (for multi-instance view) by applying Microsoft Academic Search API. For each abstract, we extract the TF-IDF (Jones, 1972) feature to construct an instance and there are 46,531 instances in all. One vs. the rest strategy is adopted to construct data sets and for each topic a same number of negative instances and positive instances are randomly selected to construct the single-instance view, then instances for corresponding references are packaged to build bags in the augmented multi-instance view. The detail of TechPaper data set is exhibited as Table 1. #obj stands for the number of origin papers. Note that it is the number of instances in the single-instance view and is also the number of bags in the augmented multi-instance view; #inst represents the total number of instances in both single-instance view and augmented multi-instance view; #inst/bag is the average number of instances per bag in the augmented multi-instance view.

4.1.2. The 5 PubMed Data Sets

We obtain the Gold standard data file of drug review topics from Cohen et al. (2006), which contains PubMed ID and Article Triage Status. Then we download abstracts of both papers (for single-instance view) and their references (for multi-instance view) according to PubMed ID. The similar to TechPaper datasets, we extract TF-IDF (Jones, 1972) feature to build both single-instance view
and multi-instance view. We construct 5 data sets in all, including ADHD, Antihistamines(AH), Estrogens(EG), OralHypoglycemics(OH) and UrinaryIncontinence(UI). Note that in PubMed Data sets, positive instances are much fewer than negative ones (the proportion of positive instances is less than 0.1). The detail of PubMed data set is exhibited as Table 2.

4.1.3. The WebPage Data Set

We reconstruct the MILWEB (Zhou et al., 2005) data to the AMIV format. The original data set contains 113 web index pages and 3,423 linked web pages, which are labeled by 9 volunteers according to their interests. Therefore, there are 9 versions. Here we also extract the TF-IDF (Jones, 1972) feature to construct an instance for each page, thus the single-instance view is composed of instances for original 113 pages and the augmented multi-instance view is composed of bags of instances for corresponding linked pages. In detail, each data set of WebPage contains 3,536 instances in all, with 113 instances in the single-instance view and 113 bags in the augmented multi-instance view. In average, each bag holds 30.29 instances.

For all above data sets, we run 10 times hold-out tests, in each run, 4/5 data are used for training and the rest 1/5 for testing. Average performance and standard deviations of the 10 runs are reported.

4.2. Compared Approaches

We compare the AMIV-lss approach with two single view learning approaches Standard and SV-mil, two variants of multi-view learning approaches Concatenation and MV-sil, one variant of multi-instance learning approach Amil, one variant of semi-supervised learning approach AlapSVM and one variant of a multi-instance multi-source approach AMI2LS. Because the experiments are held on text data sets, all of aforementioned approaches are based on linear SVM.

- **Standard**: On the only original single-instance view, a linear SVM is trained.

- **SV-mil**: On the only multi-instance view, the popular used multi-instance learning approach miSVM (Andrews et al., 2002) is applied.

- **Concatenation (Conc)**: Firstly each single view instance and its corresponding bag of instances in the augmented view is concatenated into one feature vector, then a linear SVM is trained on the new concatenated instances. Note that if the number of instances in bags is not equal, feature vectors of concatenated instances will have different length. In this case, the shorter is supplemented with 0 until the same length of feature vector is achieved.

- **MV-sil**: By taking average of instances in each bag, the augmented multi-instance view is transformed into a single instance view, thus traditional multi-view format data is obtained. Then NMF (Hsieh and Dhillon, 2011) is applied to learn a semantic subspace between views. Finally, a linear SVM is trained in that space. MV-sil approach can be regarded as the degenerated version of the AMIV-lss approach with fixed prototypes.

- **Amil**: By taking each instance in the single-instance view as a bag with only one instance and combining those special bags and bags in the augmented multi-instance view together, the original AMIV data are reformed in a MIL data format. Then miSVM (Andrews et al., 2002) is applied on the reformed MIL data.
Figure 1: Performance comparison (accuracy) on the twelve TechPaper data sets

- **AlapSVM**: On AMIV format data, regarding the instances in the single-instance view as labeled data and those in the augmented multi-instance view as unlabeled data, linear Laplacian SVM (LapSVM) (Belkin et al., 2006) is applied.

- **AMI2LS**: We adapt $M1^2LS$ (Zhang et al., 2013) approach to the AMIV framework by considering bag level consistency. We take average of a bag as its representative, then the bag level consistency constraint is that output values of classifiers on both single-view instance and its corresponding multi-instance view bag representative are similar.

For SVM tools applied in baselines and our approach, *liblinear* (Fan et al., 2008) is adopted and the parameter C is set as 1.0. In our AMIV-lss approach, for latent subspace dimension $t$, $\lambda_1(\lambda_2)$ and neighborhood size $k$, they are tuned by 5-fold cross validation in range of $[10, 20, 50]$, $[2^{−11}, 2^{−9}, \ldots, 2^{−3}]$ and $[1, 3, 5, 10]$, respectively, on training set. Besides, we set maximum iteration number as 50. For compared approaches, the other parameters are also tuned by 5-fold cross validation on training set.
Table 3: Performance comparison (F1 score) on the twelve TechPaper data sets. •(●) indicates that AMIV-lss is significantly better(worse) than the compared method (paired t-tests at 95% significance level). Bolder font indicates the best result achieved on the corresponding data set.

<table>
<thead>
<tr>
<th></th>
<th>Standard</th>
<th>SV-mil</th>
<th>Conc</th>
<th>MV-sil</th>
<th>Amil</th>
<th>AlapSVM</th>
<th>AMI2LS</th>
<th>AMIV-lss</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>.925±.035●</td>
<td>.852±.033●</td>
<td>.920±.016●</td>
<td>.935±.011●</td>
<td>.873±.035●</td>
<td>.912±.037●</td>
<td>.937±.029</td>
<td>.946±.012</td>
</tr>
<tr>
<td>CLST</td>
<td>.944±.018●</td>
<td>.881±.073●</td>
<td>.960±.018●</td>
<td>.957±.023</td>
<td>.885±.055●</td>
<td>.942±.020●</td>
<td>.960±.021</td>
<td>.961±.013</td>
</tr>
<tr>
<td>DPL</td>
<td>.914±.014</td>
<td>.833±.045●</td>
<td>.932±.022●</td>
<td>.912±.014</td>
<td>.853±.027●</td>
<td>.908±.009●</td>
<td>.912±.009</td>
<td>.916±.018</td>
</tr>
<tr>
<td>MTCL</td>
<td>.961±.014●</td>
<td>.896±.024●</td>
<td>.944±.015●</td>
<td>.971±.009●</td>
<td>.909±.019●</td>
<td>.945±.016●</td>
<td>.953±.021●</td>
<td>.981±.008</td>
</tr>
<tr>
<td>MIL</td>
<td>.854±.031●</td>
<td>.797±.028●</td>
<td>.877±.031●</td>
<td>.881±.031</td>
<td>.823±.047●</td>
<td>.839±.032●</td>
<td>.897±.014</td>
<td>.909±.021</td>
</tr>
<tr>
<td>MLL</td>
<td>.927±.026●</td>
<td>.907±.015●</td>
<td>.949±.037●</td>
<td>.943±.020</td>
<td>.917±.016●</td>
<td>.924±.018●</td>
<td>.950±.017</td>
<td>.950±.032</td>
</tr>
<tr>
<td>MTL</td>
<td>.869±.041●</td>
<td>.814±.030●</td>
<td>.871±.032●</td>
<td>.874±.057</td>
<td>.844±.044●</td>
<td>.880±.025</td>
<td>.881±.030</td>
<td>.884±.037</td>
</tr>
<tr>
<td>MVL</td>
<td>.958±.008●</td>
<td>.893±.027●</td>
<td>.941±.016●</td>
<td>.963±.008●</td>
<td>.909±.018●</td>
<td>.951±.015●</td>
<td>.938±.018●</td>
<td>.985±.008</td>
</tr>
<tr>
<td>OL</td>
<td>.979±.004●</td>
<td>.900±.028●</td>
<td>.947±.021●</td>
<td>.974±.026</td>
<td>.925±.015●</td>
<td>.970±.009●</td>
<td>.961±.018●</td>
<td>.998±.005</td>
</tr>
<tr>
<td>RL</td>
<td>.892±.022●</td>
<td>.878±.025●</td>
<td>.919±.024●</td>
<td>.873±.030</td>
<td>.891±.036●</td>
<td>.892±.022</td>
<td>.895±.023</td>
<td>.895±.030</td>
</tr>
<tr>
<td>SSL</td>
<td>.955±.023●</td>
<td>.845±.014●</td>
<td>.950±.028●</td>
<td>.963±.027</td>
<td>.880±.023●</td>
<td>.960±.029</td>
<td>.950±.014●</td>
<td>.971±.013</td>
</tr>
<tr>
<td>TL</td>
<td>.805±.029●</td>
<td>.789±.017●</td>
<td>.805±.027●</td>
<td>.818±.007</td>
<td>.809±.025●</td>
<td>.823±.030</td>
<td>.821±.033</td>
<td>.834±.012</td>
</tr>
</tbody>
</table>

Figure 2: Performance comparison (accuracy) on the five PubMed data sets

Table 4: Performance comparison (F1 score) on the five PubMed data sets. •(●) indicates that AMIV-lss is significantly better(worse) than the compared method (paired t-tests at 95% significance level). Bolder font indicates the best result achieved on the corresponding data set. – means that all the instances are classified to be negative.

<table>
<thead>
<tr>
<th></th>
<th>Standard</th>
<th>SV-mil</th>
<th>Conc</th>
<th>MV-sil</th>
<th>Amil</th>
<th>AlapSVM</th>
<th>AMI2LS</th>
<th>AMIV-lss</th>
</tr>
</thead>
<tbody>
<tr>
<td>OH</td>
<td>.228±.114●</td>
<td>.259±.018●</td>
<td>.248±.124●</td>
<td>.327±.063●</td>
<td>.330±.045●</td>
<td>.214±.120●</td>
<td>–</td>
<td>.454±.011</td>
</tr>
<tr>
<td>UI</td>
<td>.389±.053●</td>
<td>.189±.126●</td>
<td>.286±.057●</td>
<td>.553±.027</td>
<td>.493±.029●</td>
<td>.389±.053●</td>
<td>–</td>
<td>.576±.069</td>
</tr>
</tbody>
</table>
4.3. Results

Figures 1 and 2 exhibit the classification results of aforementioned approaches on abstract screening data sets, TechPaper data sets and PubMed data sets respectively. As the results indicate, our approach AMIV-lss achieves the best performance on 15 data sets and the second-best performance on the rest 2 data sets.

The Concatenation approach, concatenating two views together, tries to directly make use of augmented information, while the MV-sil approach transforms the AMIV problem into traditional multi-view learning problem which take advantage of intrinsic relationship between different views. However, as we can see, sometimes they even get worse performance than the two single view baselines. Such degenerated performance is due to the noise brought by the augmented information. AMIV-lss is more stable on the other hand, which alleviates the negative influence of noisy data by special designed key prototypes and learns the semantic representation, and outperforms Concatenation and MV-sil in most cases.

Amil which uses both single-instance view and augmented multi-instance view, always achieves better performance than SV-mil which uses multi-instance view only. However, Amil obtains worse performance than AMIV-lss in most cases, though making use of both views as well. This may be due to the fact that AMIV-lss exploit the data structure across multiple views, while Amil regard each view independently.

AlapSVM, applying LapSVM (Belkin et al., 2006) on the AMIV data, takes the instances in the single-instance view as labeled data and those in the augmented view as unlabeled data and exploits manifold structure among instances. As a result, it achieves better performance than the single view
baselines. However, AlapSVM is not the best choice for the AMIV data setting, because in the multi-instance view, instances are not only unlabeled data, we also have label information on bags. Without considering such information, AlapSVM performs worse than AMIV-lss which takes bag information into account.

AMI2LS is a variant of $MI^2LS$ (Zhang et al., 2013) approach, which is adapted to the AMIV framework by considering bag level consistency via using bag center. As bag center may hold an opposite label to the origin bag, thus to fit such bag level consistency constraint may degenerate the performance due to such noisy bag representatives. As shown in result tables, AMIV-lss outperforms AMI2LS in most cases.

Tables 3 and 4 summarize the F1 score of aforementioned approaches on abstract screening data sets. As seen, AMIV-lss outperforms other approaches on most data sets. In AMIV-lss, the prototype of a bag is the linear combination of nearest neighbors of corresponding labeled single-instance view instance in the semantic subspace with distance inverse weight, thus it is more likely to keep key information of the bag. As a result, AMIV-lss will predict more positive instances precisely.

The AMIV framework can be applied in not only abstract screening tasks, but other situations where augmented information is presented as multi-instance view bags. Figure 3 shows the results of aforementioned approaches on web page classification tasks (i.e. the WebPage Data Set), in which the augmented multi-instance bags are linked web pages. As we can see, AMIV-lss won the first prize in 5 data sets and the second prize in the rest 4. Concatenation and Amil also won the first prize for some time, but they may obtain poor performance in other cases, while AMIV-lss on the other hand is more stable with good performance.

For the convergence issue, the algorithm converges empirically: the relative difference of objective value drops below $10^{-5}$ after 10-20 iterations in most cases. For the running time issue, a coordinate descent method is applied to solve the matrix factorization problem of Eq. (6), of which in each iteration, the time complexity is approximately $O(md)$, where $m$ is the number of rows and $d$ is the number of columns. For the worst case that the algorithm stops when the maximum iteration number is reached, the algorithm scales almost linearly with respect to the number of instances with fixed $d$. Note that our work focus on prediction time, because we need to predict the label quickly when an abstract is screened, whereas the training of the prediction model can be run offline.

5. Conclusion

In this paper, we propose the AMIV (learning with Augmented Multi-Instance View) framework for tasks where a better model can be learned by exploiting a new style of augmented information presented as multi-instance bags. To the best of our knowledge, such a learning with augmented multi-instance view problem has not been touched before. Under the AMIV framework, we propose the AMIV-lss to solve this problem via establishing a latent semantic subspace between the two views. Experimental results validate the effectiveness of AMIV-lss while single view only or directly applying multi-view/multi-instance approaches get unsatisfactory results. Our work is under a single label setting, yet in real applications, an instance may possess several labels simultaneously. Therefore, it is interesting to study the AMIV framework for multi-label learning in the future.
Acknowledgments

The authors want to thank anonymous reviewers for helpful comments and suggestions, and thank W.-J. Li and Y.-F. Li for reading the draft. This research was supported by the National Key Basic Research Program of China (2014CB340501), the National Science Foundation of China (61273301, 61333014), and the Collaborative Innovation Center of Novel Software Technology and Industrialization. Z.-H. Zhou is the corresponding author of the paper.

References


