A Multi-View Two-level Classification Method for Generalized Multi-instance Problems

Xiaoguang Wang
Xuan Liu; Stan Matwin
Faculty of Computer Science
Dalhousie University, Canada
Email: {x.wang; xuan.liu}@dal.ca; stan@cs.dal.ca

Nathalie Japkowicz
School of Electrical Engineering and Computer Science
University of Ottawa, Canada
Email: nat@eecs.uottawa.ca

Hongyu Guo
National Research Council of Canada
1200 Montreal Road
Ottawa, ON., Canada
Email: hongyu.guo@nrc-cnrc.gc.ca

***Abstract*—Multi-instance (MI) learning is different than standard propositional classification, as it uses a set of bags containing many instances as input. While the instances in each bag are not labeled, the bags themselves are, as positive or negative. In this paper, we present a novel multi-view, two-level classification framework to address the generalized multi-instance problems. We first apply supervised and unsupervised learning methods to transform a MI dataset into a multi-view, single meta-instance dataset. Then we develop a multi-view learning approach that can integrate the information acquired by individual view learners on the meta-instance dataset from the previous step, and construct a final model. Our empirical studies show that the proposed method performs well compared to other popular MI learning methods.**

I. Introduction

As opposed to traditional supervised learning algorithms, multi-instance (MI) learning learns an MI dataset, which consists of bags of individual instances with unknown classifications. Only the bags are labeled, and each can contain many instances. The number of instances in each bag can be different, and the same instance can belong to different bags.

MI learning first emerged in the work of Dietterich *et al.* [1] when addressing the problem of drug activity prediction. The motivating task was to predict whether a certain molecule is active, which is determined by its shape. A molecule has different conformations because some of its internal bonds can be rotated. If at least one of the conformations of the molecule binds well to certain receptors, the molecule is considered active. In such a case, a so-called *standard MI assumption* is applied; that is, a bag is assumed to be positive if and only if it contains at least one positive instance. This assumption was introduced because it was applicable to many MI applications. Numerous algorithms that focus on identifying positive instances were subsequently developed under this assumption, including axis-parallel rectangles [1], the Diverse Density algorithm [3] and decision tree algorithm [7].

Over the past few years, several generalized views of the MI concept were also introduced [11][33]. In generalized MI problems, the learner does not follow the restriction of the *standard MI assumption*. In this paper, we address the MI problem under the *generalized MI assumption*, since it is less specific. Under this assumption, we introduce the idea of multi-view, two-level classification to deal with the generalized MI problems. The first step of this method constructs a single meta-instance from a bag, which represents regions in the instance space and has an attribute for each region. Every attribute indicates the number of instances in the bag that can be found in the corresponding region. By repeating this step using different construction methods, we get a multi-view attribute set containing attributes that represent regions in different instance spaces. Along with the bag's class label, the multi-view meta-instance can be passed to a standard propositional learner, to learn the influence of the regions on a bag's classification. In the second step, we introduce a multi-view approach to learn from the multiple independent sets of features generated in the first step. Multi-view learning describes the problem of learning from multiple independent sets of features (i.e. views) of the presented data. This framework has been successfully applied to many real-world applications [13][16][17][30]. Indeed, a multi-view learning problem with n views can be seen as n strongly uncorrelated feature sets that are distributed in the dataset.

This multi-view, two-level classification (MV-TLC) strategy transforms a multi-instance dataset into a multi-view, single meta-instance dataset, and learns from multiple views (the feature set) of this meta-dataset. The information acquired by view learners is then integrated to construct a final classification model. Our empirical studies show that our method compares well to other popular MI classifiers.

Unlike most current multi-instance learning algorithms, which are derived from supervised learning algorithms by shifting the focus from the instances to the bags (i.e. adapting single-instance algorithms to the multi-instance representation), our method demonstrates the feasibility of another approach to solving multi-instance learning: adapting the multi-instance representation to the single-instance algorithms. The main contribution of this paper is that we transform the MI problem into a multi-view single instance problem. This provides two potential benefits: any existing single instance algorithm can be applied, and the presented multi-view learning method can use the consistency among different views to achieve better performance.

The remainder of the paper is structured as follows: Section 2 presents the related concepts, Section 3 describes and discusses the proposed algorithm, and Section 4 illustrates the efficiency of our algorithm as determined by experimentation and offers final remarks and Section 5 presents the conclusion and future work.

978-1-4799-5666-1/14/$31.00 ©2014 IEEE

II. Related works

A. MI problem

Assume a binary class attribute $\Omega = \{+, -\}$ and let χ be the instance space. Then an MI concept is a function $\nu_{MI} : \mathbb{N}^{\chi} \rightarrow \Omega$. The task in MI learning is to learn this function based on a number of example elements of the function. Here, $\mathbb{N}^{\chi}$ refers to the set of all functions from χ to $\mathbb{N}$ (which is isomorphic to the set of all multi-subsets of χ). The output of $f(\chi) \in \mathbb{N}^{\chi}$ is viewed as the number of occurrences of χ in the multi-set. Weidmann *et al.* [11][24] indicated that by employing different assumptions of how the instances'classifications determine their bag's label, different kinds of MI problems can be defined. Under the standard MI assumption, the MI learning problem can be defined as:

$$v_{MI}(X) \Leftrightarrow \exists x \in X : c_i(x) \tag{1}$$

The *standard MI assumption* states that each instance has a hidden class label $c \in \Omega = \{+, -\}$. Under this assumption, an example is positive only if one or more of its instances are positive. Thus, the bag-level class label is determined by the disjunction of the instance-level class labels.

Although the *standard MI assumption* is widely believed to be appropriate for the musk drug activity prediction problem [1], the MI representation can be applied to a number of other problem domains where the *standard MI assumption* might not be directly applicable. Based on this, Weidmann *et al.* [11] formulated a hierarchy of *generalized instance-based assumptions* for MI learning. The hierarchy consists of the *standard MI assumption* and three types of *generalized MI assumptions*: presence-based MI, threshold-based MI and count-based MI, each more general than the previous.

The formal definitions of presence-based MI, threshold-based MI and count-based MI are shown in Equations 2, 3 and 4.

$$v_{PB}(X) \Leftrightarrow \forall c \in \hat{C} : \Delta(X, c) \geq 1 \tag{2}$$

$$v_{TB}(X) \Leftrightarrow \forall c_i \in \hat{C} : \Delta(X, c_i) \geq t_i \tag{3}$$

$$v_{CB}(X) \Leftrightarrow \forall c_i \in \hat{C} : t_i \leq \Delta(X, c_i) \leq z_i \tag{4}$$

In these equations, $\nu_{PB} : \mathbb{N}^{\chi} \rightarrow \Omega$ defines a presence-based MI concept, $\nu_{TB} : \mathbb{N}^{\chi} \rightarrow \Omega$ defines a threshold-based MI concept, and $\nu_{CB} : \mathbb{N}^{\chi} \rightarrow \Omega$ defines a count-based MI concept. $\hat{C} \in C$ is the set of required concepts, $\Delta : \mathbb{N}^{\chi} \times C \rightarrow \Omega$ is the function that outputs the number of occurrences of a concept in the bag, $t_i \in \mathbb{N}$ is the lower threshold for concept I, and $z_i \in \mathbb{N}$ is the upper threshold for concept I. We also have $\nu_{MI} \subset \nu_{PB} \subset \nu_{TB} \subset \nu_{CB}$.

For our approach to be used in wider domains, algorithms that rely on the *generalized MI assumption* may be more appropriate.

While MI learning has been used in many applications, including drug activity recognition [1], text-categorization [9] and computer vision recognition [7], there is a great deal of research about, and many different approaches to, solving the MI learning problem.

Among these solutions to MI learning problems, one approach has become increasingly popular: upgrading paradigms from normal single-instance learning to handle MI data. For example, Diverse Density (DD) [3] and the Expectation-Maximization version [5] were proposed as general frameworks for solving multi-instance learning problems. The k Nearest Neighbour approach, known as Citation kNN, was adapted for MIL problems in [4]. Andrews *et al.* [6] proposed two approaches to modify Support Vector Machines: mi-SVM for instance-level classification, and MI-SVM for bag-level classification. For tree methods, Blockeel *et al.* [7] proposed a multi-instance tree method (MITI), and Bjerring *et al.* [9] extended this by adopting MITI to learn rules (MIRI).

The two-level learning approach has also become increasingly popular. This approach extracts distributional properties from each bag of instances for each class, and attempts to discriminate classes (or groups) according to their distributional properties. The first step is to collect all the instances from all the bags, and label each with the label of the bag they come from. This effectively creates a propositional (i.e. single-instance) dataset. Any propositional learning method can be used on this dataset to learn the instance level distribution. The second step of this approach is to discriminate classes in the bag level base on the achieved instance level distribution. Xu [33] investigated a simple heuristic algorithm (MIWrapper) to apply single-instance learners under the collective assumption, which is an alternative to the generalized MI assumption. Weidmann *et al.* [11] presented a Two-Level Classification (TLC) algorithm to learn the type of MI concepts that are described in their concept hierarchy. Similarly, Zhou and Zhang [12] presented a constructive clustering ensemble (CCE) method. This algorithm uses a clustering method to group the instances in the training bags into d clusters, to build the concepts for the bag level classification.

B. Multi-view learning

In recent years, numerous methods to learn from multi-view data by considering the diversity of different views have been proposed. The views can be obtained from multiple sources or different feature subsets. The basic idea of Multi-view learning is to make use of the consistency among different views to achieve better performance. As opposed to single view learning, multi-view introduces one function to model a particular view, and jointly optimizes all the functions to exploit the redundant views of the same input data to improve the learning performance. The Multi-view learning process has three stages: Multi-view construction, views validation and views combination.

Multi-view construction methods can be analyzed and categorized into three classes. The first class includes techniques that construct multiple views from the so-called meta-data, using random approaches. Creating different views corresponds to feature set partitioning, which generalizes the task of feature selection. Instead of providing a single representative set of features, feature set partitioning decomposes the original set into multiple disjoint subsets to construct each view. For example, Brefeld *et al.* [16][17] presented a simple way to convert from a single view to multiple views, by splitting the original feature set into different views at random. Di and Crawford [21] conducted a thorough investigation of view generation for hyper-spectral image data. Three strategies: Clustering, Random selection and Uniform band slicing have

been proposed to construct multiple views by considering the key issues of diversity, compatibility and accuracy.

The second class consists of algorithms that reshape or decompose the original single-view feature into multiple views, such as the above matrix representations, or different kernel functions. For example, Wang *et al.* [21] developed a novel technique to reshape the original vector representation of a single view into multiple matrix representations. Moreover, the problem of how to learn the kernel combination can be seen as multiple kernel learning [23].

The third class is comprised of methods that automatically perform feature set partitioning. Chen *et al.* [22] suggested a novel feature decomposition algorithm called Pseudo Multi-view Co-training (PMC), which can find an optimal split of the features automatically, by solving an introduced optimization problem iteratively.

With respect to views validation, several approaches have been proposed to analyze the relationships between multiple views, or to cope with the problems resulting from the violation of view assumptions or the noise in the views. Muslea *et al.* [24] introduced a view validation algorithm that predicts whether the views are adequately compatible to solve multi-view learning tasks. Liu and Yuen [25] proposed two new confidence measures (inter-view confidence and intra-view confidence) to describe the view sufficiency and view dependency issues in multi-view learning. For multiple kernel learning, Lewis *et al.* [26] compared the performance of unweighted and weighted sums of kernels on a gene functional classification task.

Regarding views combination, Kumar and Daume III [27] applied co-training to the unsupervised learning setting, and proposed a spectral clustering algorithm for multi-view data. In the Bayesian co-training proposed by Yu *et al.* [28], a Bayesian undirected graphical model for co-training through the gauss process is constructed. For multiple kernel learning, Gonen and Alpaydin [23] proposed assigning different weights to kernel functions according to data distribution, and defined a locally combined kernel matrix.

III. Proposed Methods

The MI concepts described by Weidmann *et al.* [11] consist of a set of instance-level processes that are in some way related to the bag-level concepts. According to the definition of standard and generalized multi-instance learning, the label of a bag is determined by the relationship between the feature vector set describing the bag and the target points in the instance space. There are two functions in the Weidmann *et al.* [11] concept hierarchy that determine a bag's class label: the mono-instance concept function that assigns an instance in a bag to a concept, and the MI concept function that computes a class label from the instances in a bag, given their concept membership by the first function. Thus, a two-level approach to learning is appropriate.

Most current MI learning algorithms use a strategy of adapting single-instance learning algorithms to meet the MI representation, and have been somewhat successful. And some MI learning algorithms using the opposite strategy, that is, adapting the MI representation to meet the requirements of existing single-instance supervised learning algorithms, have also been effective [11][12][33].

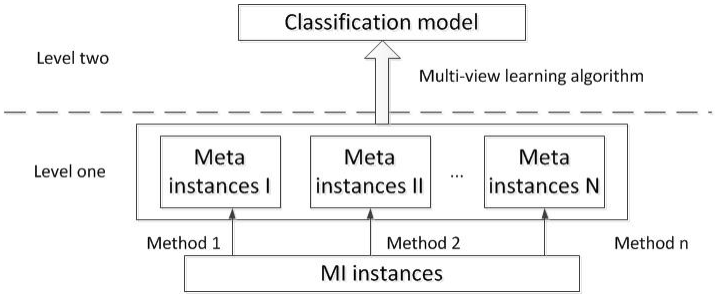

Fig. 1. MV-TLC framework.

The TLC approach proposed by Weidmann *et al.* generates only one meta-instance for each bag, using a standard decision tree. In contrast, CCE proposed by Zhou and Zhang [12] employs k-means with a different number of clustered groups to impose different structures on the instance space; in each structure a meta-instance is generated for each bag. CCE also uses majority voting to combine the predictions of classifiers. It is clear that the role of the decision tree in TLC can be replaced by another supervised learning algorithm, and the role of k-means in CCE can be replaced by some other clustering method. This provides a starting point to develop an upgrading methodology based on TLC and CCE.

In this paper, we present a Multi-view two-level classification framework (MV-TLC) for generalized multi-instance learning, as shown in Figure 1. In the first level, the framework imposes different structures on instance spaces with different dimensions, and in each structure a meta-instance is generated for each bag. Both the supervised and unsupervised learning algorithms are employed to generate the meta-instances. In the second level, a multi-view algorithm is applied to combine the predictions of the classifiers.

A. Construct the multi-view meta-instances dataset in level one

In level one, the instances in all the bags are collected, supervised and unsupervised learning methods are then applied to construct a new concept for the second level. The same mapping is performed at classification time, and the bag-level predictions are made by the single-instance learner.

1) Using supervised learning methods to construct the meta-instances: Several algorithms can be used for the supervised learning methods, including decision tree (which is also used in TLC [11]) and rule induction. Here we choose decision tree to impose a structure on the instance space. The decision tree is built on the set of all instances contained in all bags, and labeled with their bag's class label. A unique identifier is assigned to each node or leaf of the tree, and information gain is used for test selection. A simple pre-pruning heuristic is applied, and nodes are not split further when the sum of the instance weights in the node is less than two. Each node in the tree represents a concept. Algorithm 1 illustrates the process.

In this algorithm, each bag is converted into a single-instance representation with an attribute for every node in the tree (i.e. each concept), the value of which is set to the

Algorithm 1 TLC-decision tree

Given: D = the set of train bags; C = all instances in the bags in D
Set: L = decision tree classifier; F = a new single instance data set; $int\ i, j = 0$
for all $C_i \in C$ **do**
    $C_i.setClassValue(D_{(C_i)}).ClassValue)$
end for
$L.train(C)$
Output the final hypothesis:
set N = all nodes and leaves in L
while $j < D.size()$ **do**
    $F_j.setClassValue(D_j)$
    $F_j.setAttribute(N)$
    for $n \in 1, N.size()$ **do**
        $F_j.attribute(n) = \sum count(N_{(C_i \in D_j)})$
    end for
end while
Return F

number of instances that reach that node in the decision tree. The tree allows us to convert a bag into a single instance, with one numerical attribute for each node in the tree. Each attribute counts how many instances in the bag are assigned to the corresponding node in the tree. The TLC-decision tree algorithm proposed in algorithm 1 is similar to the method used in TLC [11], but there are some important differences. For instance, we do not initialize the weight of each instance. Moreover, unlike TLC, which only counts nodes as attributes, in our method both the leaves and the nodes are utilized. Although only a TLC-decision tree method is proposed here, other supervised learning algorithms can also be used to construct new concepts in level one, such as rule induction algorithms and alternative tree methods.

2) Using unsupervised learning methods to construct the meta-instances: For the unsupervised learning methods, clustering algorithms are chosen to construct new concepts, and then applied to cluster the instances into d groups. Specifically, d features are generated so that if a bag has an instance in the i-th group, then the value of the i-th feature is set to 1; otherwise it is set to 0. Thus, each bag is represented by a d-dimensional binary feature vector, such that common single-instance supervised classifiers can be employed to distinguish the bags. Theoretically, any unsupervised learning algorithm can be used in the proposed TLC-clustering algorithm (e.g. k-means, EM). Algorithm 2 illustrates the process.

B. Learning the multi-view meta-instances dataset in the level two

Combining the generated concepts in level one gives us a multi-view single instance meta-dataset for level two. To learn this dataset, we introduce a Correlation-based Multi-View (CMV) algorithm in level two of MV-TLC. Level two of MV-TLC includes three stages: a multiple views construction stage, a view validation stage and a view combination stage. Algorithm 3 provides the details of the CMV algorithm.

1) Multi-view construction and validation: In the second level of the MV-TLC framework, the Multiple Views Construction Stage builds various hypotheses on the target concept,

Algorithm 2 TLC-clustering

Given: D = the set of train bags X^l; C = all instances in the bags in D
Set: T = a set of m numbers $t_1, t_2, \cdots, t_m$; U = Clustering algorithm; S_i = a new single instance data set
int $i, j = 0$
for $i \in 1, \cdots, m$ **do**
    $U(C, t_i)$
    for $j \in 1, \cdots, D.size()$ **do**
        for $k \in 1, \cdots, t_i$ **do**
            $y_k^j \Leftarrow Overlap(X^j, U_k)$
        end for
        $S_i \Leftarrow S_i \bigcup \{< y_1^j, \cdots, y_{t_m}^j >\}$
    end for
    $S_i.setClassValue(D)$
end for
Return S_i

Algorithm 3 Correlation-based Multi-view learning algorithm

Given: dataset Φ which is generated from level one of MV-TLC
Set: L = view learner; M = meta learner
Output: the final hypothesis: $\pounds$
Let View Set $V = \emptyset$; Hypothesis set $H = \emptyset$
Generate view set $V = V^1, \cdots, V^m$ using correlation-based Feature selector and measurement described in equation (5) and (6) from Φ
Train L with V, forming hypothesis set H
Form final model F by combining H, using M
Return $\pounds$

based on the multiple training data sets given by level one of the MV-TLC. Conventional single-table data mining methods (view learners) are used to learn the target concept from each view of the database separately.

In the views validation stage, a number of different view learners are trained. All the learners from the views construction stage are evaluated in this stage, as they must be validated before being used by the meta-learner. This process is required to ensure they are capable of learning the target concept on their respective training sets. In addition, strongly uncorrelated view learners are preferred.

The goal of the CMV algorithm is to select a subset of views that are highly correlated with the target concept, but irrelevant to one another. As stated, this is an ideal assumption, since one rarely encounters real world problems with independent views. The algorithm uses a heuristic measure to evaluate the correlation between views; a similar heuristic principle was applied in the feature selection approach by Hall [14].

To generate different views, a correlation-based feature selector is chosen in our algorithm:

$$r_{zc} = \frac{k\overline{r_{zl}}}{\sqrt{k + k(k-1)\overline{r_{ll}}}} \tag{5}$$

where r_{zc} is the correlation between the summed components and the outside variable, k is the number of components, $\overline{r_{zl}}$ is the average of the correlations between the components

and the outside variable, and $\overline{r_{ll}}$ is the average inter-correlation between components.

To measure the correlation between features, Symmetrical Uncertainty [14] is used to calculate $\overline{r_{zl}}$ and $\overline{r_{ll}}$. The Symmetrical Uncertainty S is defined as follows:

$$S = 2.0 \times [\frac{Gain}{H(Y) + H(X)}] \tag{6}$$

where $Gain = H(Y) + H(X) - H(X,Y)$ and $H(Y) = -\sum_{y \in Y} p(y) \log_2 p(y)$.

Furthermore, Symmetrical Uncertainty compensates for information gain bias toward attributes with more values, and normalizes its value to the range [0,1].

After completing construction of the view feature set, the CMV algorithm ranks view feature subsets according to the correlation-based heuristic evaluation measure S. It searches all possible view feature subsets and ranks them, then selects the best ranking subset. Various heuristic approaches are employed to search the view feature space. The CMV method uses best first and ranker [14] as search strategies.

2) Multi-view combination: In the last step of the MV-TLC strategy, the multi-view learners from the Views Validation Stage are incorporated into a meta-learner, to construct the final classification model. As shown in algorithm 3, the meta-learner is used to create a function to control how the view learners work together to achieve maximum classification accuracy. This function, and the hypotheses constructed by each of the view learners, constitutes the final model.

Since multi-view learners can result in various performances and it is difficult to guarantee their comparable performance, the meta-learning method is very suitable to the multi-view learning framework.

After constructing the multi-view training data sets, the multi-view algorithm calls on the view learners to learn the target concepts from the sets. Each learner constructs a different hypothesis, based on the data it receives. In this step any traditional single-table learning algorithms can be applied. In this way, all view learners make different observations on the target concept, based on their perspective. A meta-leaner is trained using the constructed meta-data in order to achieve strong predictive performance. The results from the learners is then validated and combined to construct the final classification model.

C. Analysis

Blum and Mitchell [13] proved that when two adequate views are conditionally independent given the class label, co-training can be successful. Based on the same conditional independence assumption, Dasgupta *et al.* [29] provided the PAC style bounds for co-training. Let S be an $i.i.d$ sample consisting of individual samples $s_1, \cdots, s_m$. A partial rule h on a dataset X is a mapping from X to the label set $\{1, \cdots, k, \perp\}$, where k is the number of class labels and $\perp$ denotes the partial rule h that gives no opinion. We have the following for all pairs of rules h_1 and h_2: If $\gamma_i(h_1, h_2, \delta/2) > 0$ for $1 \leq i \leq k$ then f is a permutation, and for all $1 \leq i \leq k$, we have:

$$P(h_1 = i | f(y) = i, h_1 \neq \perp) \leq \frac{1}{\gamma_i(h_1, h_2, \delta)} (\epsilon_i(h_1, h_2, \delta) + \hat{P}(h_1 \neq i | h_2 = i, h_1 \neq \perp)) \tag{7}$$

where $\epsilon_i(h_1, h_2, \delta) = \sqrt{\frac{\ln 2(|h_1|+|h_2|)+\ln 2/\delta}{2|S(h_2=i, h_1 \neq \perp)|}}$ and $\gamma_i(h_1, h_2, \delta) = \hat{P}(h_1 = i | h_2 = i, h_1 \neq \perp) - \hat{P}(h_1 \neq i | h_2 = i, h_1 \neq \perp) - 2\epsilon_i(h_1, h_2, \delta)$. Abney [32] relaxed this assumption, and found that weak dependence alone can lead to successful co-training. Given the mapping function $Y = y$, the conditional dependence of opposing-view rules h_1 and h_2 is defined as: $d_y = \frac{1}{2} \sum_{u,v} |Pr[h_1 = v | Y = y, h_2 = u] - |Pr[h_1 = v | Y = y]|$. And the weak dependence rule is :

$$d_y \leq p_2 \frac{q_1 - p_1}{2 p_1 q_1} \tag{8}$$

where $p_1 = \min_u Pr[h_2 = u | Y = y]$, $p_2 = \min_u Pr[h_1 = u | Y = y]$, and $q_1 = 1 - p_1$. As the proposed MV-TLC framework deploys multi-views constructed using supervised or unsupervised learning methods, h_1 and h_2 can be learned from view concepts constructed using different methods. In this case, X_1 and X_2 are constructed in different feature spaces, and we can consider that the weak dependence rule in equation 8, or the conditional independence rule in equation 7, are satisfied. Under these assumptions, the CMV algorithm will be successful.

However, X_1 and X_2 can be constructed using the same method with different or identical parameter settings, and in these situations additional weak assumptions are required to guarantee the success of the CMV algorithm. We can use the ϵ-expansion assumption, first mentioned by Balcan *et al.* [30], to analyze the success condition. Here, we provide the definition of ϵ-expansion. We assume that the examples in algorithm 3 are drawn from a distribution D over an instance space X, and let X^+ and X^- denote the positive and negative regions of X respectively. For $S_1 \subseteq X_1$ and $S_2 \subseteq X_2$ let $S_i (i = 1, 2)$ denote the event of an example where $\langle x_1, x_2 \rangle$ has $x_i \in S_i$. If we set S_1 and S_2 as confident sets in each view, then $Pr(S_1 \wedge S_2)$ denotes the probability mass of examples for which we are confident about both views, and $Pr(S_1 \oplus S_2)$ denotes the probability mass of examples for which we are confident about only one view. D^+ is ϵ-expanding if, for any $S_1 \subseteq X_1^+$ and $S_2 \subseteq X_2^+$, $Pr(S_1 \oplus S_2) \geq \epsilon \min[Pr(S_1 \wedge S_2), Pr(\overline{S_1} \wedge \overline{S_2})$. Another slightly more powerful type of expansion, known as 'left-right expansion', can be defined as: D^+ is ϵ-right-expanding if, for any $S_1 \subseteq X_1^+$ and $S_2 \subseteq X_2^+$, $if\ Pr(S_1) \leq 0.5\ and\ Pr(S_2|S_1) \geq 1 - \epsilon\ then\ Pr(S_2) \geq (1 + \epsilon) Pr(S_1)$. If the MI problem is under the *standard MI assumption*, and the learning algorithm used in each view is naturally confident about being positive and is able to learn from positive examples only, then the assumption of ϵ-expanding will be satisfied, and we can say that the distribution D^+ over positive examples is expanding.

If the MI problem is under the generalized MI assumption, many concept classes cannot be learned from positive examples only. In this situation, Wang and Zhou [31] demonstrated

TABLE I. DETAILS OF DATASETS ('#'DENOTES 'NUMBER OF').

Dataset	#bags	# attribute	# positive	# negative	instances
Elephant	200	230	100	100	1391
Fox	200	230	100	100	1320
Tiger	200	230	100	100	1220
M_atoms	188	10	125	63	1618
M_bonds	188	16	125	63	3995
M_chains	188	24	125	63	5349
Musk1	92	166	47	45	476
Musk2	102	166	39	63	6598

that when the diversity between two learners is larger than their errors, the performance of the learners can be improved by multi-view style algorithms. The difference $d(h_i, h_j)$ between the two classifiers h_i and h_j implies their different biases. In the meta-level of algorithm 3, if the examples labeled by classifier h_i are useful for classifier h_j, h_i should have information that h_j does not have; in other words, h_i and h_j should have significant differences. In the CMV algorithm, we chose different meta-learners to guarantee the differences between h_i and h_j.

IV. EXPERIMENTAL STUDY

In this section, we explain our experiments to investigate and compare the proposed MV-TLC method to other popular MI learning algorithms.

A. Details of Datasets

The datasets used in our experiments are those employed in [8] and [9], and can be retrieved from http://www.eecs.uottawa.ca/~bwang009/. Table 1 shows the details of the datasets.

B. Experimental results

In our experiments, we compared MV-TLC with other well known MI algorithms. The compared algorithms with assigned classifier numbers include (1)MITI [7], (2)MILR [33], (3)DD [3], (4)EMDD [5], (5)MISMO (RBF kernel) [6], (6)SimpleMI [33] with RF, (7)SimpleMI with SVM(RBF), (8)MIwrapper [33] with RF, (9)MIwrapper with SVM(RBF), (10)MIBoost [33] with RF, (11)MIBoost with SVM(RBF), (12)TLC with attribute selection (TLC_AS) [11] and (13)MV-TLC.

Here "RF" and "SVM(RBF)" denote that Random Forest [19] and SVM with RBF kernel were chosen as the propositional learner. For MV_TLC, in level one we chose a standard decision tree method as the supervised learning method, with k-means using Euclidean distance and k-means using Manhattan distance as unsupervised learning methods to generate the concepts. Empirically, each clustering method groups the instances into 20 groups. In level two, SVM (RBF kernel), SVM (linear kernel) and Random Forest were chosen as the views learners, and SVM (RBF kernel) was chosen as the propositional learner for meta-instances. We choose both Accuracy and AUC as the measurements for our algorithms and experiments. Tables 2 and 3 present the experimental results using accuracy and AUC separately.

We applied a statistical test method–Nemenyi's post-hoc test [10] to determine which classifier had the best performance. First, we ranked the evaluation values for each dataset with different classifiers, and the sum of the ranks for all datasets is represented by $R_{.i}$, where i represents a classifier. Then we used the following formula to calculate the q value between different classifiers:

$$q_{ij} = (\overline{R_i} - \overline{R_j}) / \sqrt{\frac{k(k+1)}{6n}} \tag{9}$$

where k is the number of classifiers and n is the number of datasets. We then determined if one algorithm is better than another by comparing their q values with the critical value q_α. The results are shown in Tables 2 and 3 respectively(with column name "result"). In detail, the result of 1-8-3 for classifier 2 means that the algorithm wins once, loses eight times, and equal three times compared to other classifiers. If we set the scores as win=1, lose=-1, and equal=0, the total score of each algorithm in these two tables were calculated and summed. The final scores were shown in the last column of Table 3 (with column name "S").

Tables 2 and 3 show that the performance of MV-TLC is as good as MI-wrapper with Random Forest(both of them get score 22) and both of them are better than the other algorithms. Although MV-TLC is not designed for standard MI problems, its performance is comparable to other algorithms which achieve the best results.

C. Discussion

Data with complex structures, such as MI data, is usually difficult to learn with traditional machine learning paradigms. Constructive induction is a general approach to address inadequate features found in original data. Using this strategy, TLC and CCE shows good performance compared to other popular MI algorithms. However, the fact that TLC only generates the concepts in one instance space can make the constructed features not adequate enough to be learned. Although CCE employs clustering to impose different structures, and uses the power of ensemble learning rather than single classifier, its performance on count-based MI is as good as that of TLC with AS or MI kernel [12]. Zhou and Zhang [12] found that this could be because the binary feature vectors used by CCE are not sufficient to represent the exact number of instances in a cluster. To use the power of ensemble learning, Dietterich [2] indicated that an effective paradigm for generating diverse classifiers is required. Therefore the feature vector diversity generated by clustering the instances into different number of groups, may not be strong enough for the ensemble learning method.

The empirical study shows that by using a multi-view style algorithm, and generating multi-view concepts with different structures in different instance spaces, MV-TLC can achieve higher and more stable performance compared to TLC and CCE.

Under the *collective assumption* that instances contribute equally and independently to bag-level class labels, MI-wrapper with Random Forest showed great performance. However, for some applications this assumption might not always be suitable. In data with more complex construction, instances could be dependent to bag-level class labels, and in such cases the *generalized MI assumption* utilized by MV-TLC may be more appropriate. On the other hand, we found that for some data, both MI-wrapper and MIboost are sensitive to propositional learners. In our empirical study, when choosing SVM

TABLE II. EXPERIMENTAL RESULTS(ACCURACY BY PERCENT)

NO.	Elephant	Fox	Tiger	M_atoms	M_bonds	M_chains	Musk1	Musk2	Result
MITI	77.7±1.0	61.4±3.7	75.9±2.0	80.4±2.8	79.6±2.6	82.0±1.0	70.7±2.8	70.0±1.1	5-5-2
MILR	75.4±1.4	60.2±5.0	75.3±2.9	72.0±1.0	74.6±2.2	76.2±0.8	70.9±1.6	76.5±2.1	1-8-3
DD	75.7±1.4	65.4±1.6	66.5±2.7	71.6±0.5	71.6±1.5	75.2±1.9	77.2±4.1	76.9±2.5	2-7-3
EMDD	74.3±3.4	62.2±0.4	72.5±1.5	72.2±1.2	72.1±2.6	69.7±4.4	82.2±2.7	84.7±2.0	4-6-2
MISMO(RBF)	82.3±2.0	55.1±3.3	81.7±0.8	68.3±0.7	81.5±1.5	83.5±2.0	86.3±2.0	82.5±1.8	8-3-1
SimpleMI with RF	79.4±1.9	59.5±4.5	77.3±2.0	79.3±2.4	85.2±2.8	82.4±1.1	77.6±3.6	78.0±3.4	6-5-1
SimpleMI with SVM(RBF)	84.0±1.1	59.7±2.3	79.7±1.4	68.0±0.2	69.0±1.7	75.1±1.2	51.1±0.0	61.8±0.0	2-8-2
MIwrapper with RF	85.7±1.0	62.8±2.4	81.7±1.2	82.6±1.8	80.2±2.0	82.9±2.0	85.0±3.0	78.2±1.6	11-0-1
MIwrapper with SVM(RBF)	82.5±0.9	60.4±1.3	78.0±1.1	66.5±0.0	66.8±0.3	67.0±0.0	49.1±0.5	61.8±0.0	0-11-1
MIBoost with RF	84.5±1.1	60.6±1.8	79.1±2.1	80.6±1.4	80.0±0.9	81.7±1.1	85.0±4.4	78.4±1.6	9-2-1
MIBoost with SVM(RBF)	83.1±1.6	60.5±1.7	78.4±1.1	66.5±0.0	66.9±0.2	67.0±0.0	48.9±0.0	61.8±0.0	0-10-2
TLC_AS	81.5±3.1	63.5±3.4	75.0±2.2	78.2±1.6	82.4±0.7	83.8±0.6	84.8±1.9	81.4±1.7	8-2-2
MV-TLC	82.5±2.7	68.0±1.5	79.0±1.9	79.7±2.1	86.2±1.2	84.2±0.8	84.8±2.9	83.3±1.1	11-0-1

TABLE III. EXPERIMENTAL RESULTS(AUC)

NO.	Elephant	Fox	Tiger	M_atoms	M_bonds	M_chains	Musk1	Musk2	Result	S
MITI	77.5±1.3	61.3±2.7	75.7±1.8	73.5±3.7	72.3±3.4	74.4±1.7	70.4±2.7	71.7±1.4	0-11-1	-11
MILR	81.8±3.5	56.1±5.4	79.9±2.7	77.9±0.5	80.6±3.7	80.8±2.7	74.3±1.5	83.2±1.5	3-4-4	-8
DD	83.8±1.0	69.3±0.7	72.2±4.6	75.7±0.4	76.0±0.6	81.5±0.8	83.0±7.0	76.9±2.5	3-4-5	-6
EMDD	80.4±4.5	63.2±2.0	76.3±1.6	64.6±3.7	73.3±6.6	72.8±5.7	85.7±6.3	90.6±3.2	2-9-1	-9
MISMO(RBF)	82.3±2.0	55.1±3.3	81.7±0.8	60.0±1.3	79.9±2.2	81.3±2.7	86.2±2.0	82.4±2.3	2-5-5	2
SimpleMI with RF	88.1±2.2	63.2±3.6	85.0±2.0	83.6±1.8	90.5±1.8	85.1±0.9	86.1±2.5	85.5±3.2	9-3-0	7
SimpleMI with SVM(RBF)	84.0±1.1	59.7±2.3	79.7±1.4	53.9±0.4	56.9±1.5	65.1±1.3	50.0±0.0	50.0±0.0	0-11-1	-17
MIwrapper with RF	93.6±1.1	67.9±1.2	88.7±1.0	84.7±1.5	82.9±2.4	85.9±0.8	92.9±2.0	82.9±3.8	11-0-1	22
MIwrapper with SVM(RBF)	88.4±0.9	66.1±2.7	85.5±1.0	74.9±0.0	79.9±2.1	75.8±2.6	51.9±0.6	53.9±0.0	3-4-5	-12
MIBoost with RF	92.3±1.1	66.2±1.2	87.9±1.3	84.4±0.9	83.3±0.4	85.9±1.2	90.8±2.2	81.7±2.2	10-2-0	15
MIBoost with SVM(RBF)	88.3±1.5	64.4±1.3	85.8±1.2	73.5±0.0	78.7±3.4	77.6±0.8	53.4±0.0	55.0±0.0	3-4-5	-11
TLC_AS	81.4±2.3	62.0±2.1	77.0±1.4	76.7±1.2	83.9±2.0	84.9±1.8	83.8±2.3	82.3±1.4	4-4-4	6
MV-TLC	91.4±1.7	69.0±1.5	81.9±1.9	86.4±1.3	91.2±0.6	89.8±0.9	90.9±2.4	87.2±2.3	11-0-1	22

with RBF kernel as the propositional learner, MI-wrapper and MIboost's performance is low. In contrast, MV-TLC can employ the power of multi-view learning, and is not sensitive to propositional learners.

V. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we proposed a framework that demonstrates a solution for multi-instance learning, by adapting multi-instance representation to single-instance algorithms. The proposed algorithm, known as MV-TLC, employs processes that include both supervised and unsupervised learning methods; to help construct new multi-view meta-features that can be exploited by common supervised learning algorithms. MV-TLC also utilizes the power of multi-view learning paradigms to achieve strong generalization ability. Experiments show that MV-TLC works well with both standard and generalized multi-instance problems, without requiring any modification. There are many potential ways to modify multi-instance representation. Exploring other schemes for adapting it to single-instance algorithms would be an interesting future direction. Moreover, the success of MV-TLC reveals that other multi-view methods can be considered when learning data with complex structures. Working to apply these techniques would be another promising area for further investigation.

REFERENCES

[1] Dietterich, T., Lathrop, R., Lozano-Perez, T.: Solving the multiple instance problem with the axis-parallel rectangles. In: Artificial Intelligence, 89(1-2), 3171 (1997)

[2] Dietterich T.: Ensemble methods in machine learning. In Kittler J, Roli F (eds). Lecture Notes in Computer Science 1867, Springer, Berlin, pp 1-15(2000)

[3] Maron, O., Lozano-Prez T.: A framework for multiple instance learning. In: Proc. of the 1997 Conf. on Advances in Neural Information Processing Systems 10, p.570-576(1998)

[4] Wang, J., Zucker, J.D.: Solving the multiple-instance problem: A lazy learning approach. In: ICML (2000)

[5] Zhang, M.L., Goldman, S.: Em-dd: An improved multi-instance learning technique. In: NIPS (2002)

[6] Andrews, S., Tsochandaridis, I., Hofman, T.: Support vector machines for multiple instance learning. In: Adv. Neural. Inf. Process. Syst. 15, 561568 (2003)

[7] Blockeel, H., Page, D., Srinivasan, A.: Multi-instance tree learning. In: ICML (2005)

[8] Foulds, J., Frank, E.: Revisiting multiple-instance learning via embedded instance selection. In: W. Wobcke & M.Zhang(Eds), 21st Australasian Joint Conference on Artificial Intelligence Auckland, New Zealand, (pp. 300-310)(2008)

[9] Bjerring, L., Frank, E.: Beyond trees: Adopting MITI to learn rules and ensemble classifiers for multi-instance data. In: D. Wang & M. Reynolds (Eds.), AI 2011, LNAI 7106 (pp. 41-50)(2011)

[10] Japkowicz, N., Shah, M.: Evaluating Learning Algorithms: A Classification Perspective. Cambridge University Press (2011)

[11] Weidmann, N., Frank, E., and Pfahringer, B. (2003). A two-level learning method for generalized multi-instance problems. In Machine Learning: ECML 2003, Lecture Notes in Computer Science, pages 468479. SpringerBerlin.

[12] Zhou, Z.-H. & Zhang, M.-L. 2007. Solving multi-instance problems with classifier ensemble based on constructive clustering. Knowledge and Information Systems 11(2), 155170.

[13] Blum, A. and T. Mitchell: 1998, Combining Labeled and Unlabeled Data with Co-training.In: Proceedings of the Workshop on Computational Learning Theory.

[14] Hall, M.: 1998, Correlation-based Feature Selection for Machine Learning, PH.D diss., Waikato Uni..

[15] Bickel, S. and Scheffer,T.: Multi-view clustering. In Proceedings of the IEEE international conference on data mining, volume 36, 2004.

[16] U. Brefeld and T. Scheffer. Co-em support vector learning. In Proceedings of the twenty-firstinternational conference on Machine learning, page 16. ACM, 2004.

[17] U. Brefeld, C. Buscher, and T. Scheffer. Multi-view discriminative sequential learning.Machine Learning: ECML 2005, pages 6071, 2005.

[18] U. Brefeld, T. Gartner, T. Scheffer, and S. Wrobel. Efficient co-

regularised least squares regression. In Proceedings of the 23rd international conference on Machine learning, pages 137144. ACM, 2006.

[19] T.K. Ho. The random subspace method for constructing decision forests. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20(8):832844, 1998.

[20] W. Di and M.M. Crawford. View generation for multiview maximum disagreement basedactive learning for hyperspectral image classification. Geoscience and Remote Sensing,IEEE Transactions on, (99):113, 2012.

[21] Z. Wang, S. Chen, and D. Gao. A novel multi-view learning developed from single-view patterns. Pattern Recognition, 2011.

[22] M. Chen, K.Q. Weinberger, and Y. Chen. Automatic feature decomposition for single view co-training. In International Conference on Machine Learning, 2011.

[23] M. Gonen and E. Alpaydn. Multiple kernel learning algorithms. Journal of Machine Learning Research, 12:22112268, 2011.

[24] I. Muslea, S. Minton, and C.A. Knoblock. Adaptive view validation: A first step towards automatic view detection. In MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-, pages 443450. Citeseer, 2002b.

[25] C. Liu and P.C. Yuen. A boosted co-training algorithm for human action recognition. IEEE transactions on circuits and systems for video technology, 21(9):12031213, 2011.

[26] D.P. Lewis, T. Jebara, and W.S. Noble. Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure. Bioinformatics, 22(22):27532760, 2006.

[27] A. Kumar and H. Daume III. A co-training approach for multi-view spectral clustering. In International Conference on Machine Learning, 2011.

[28] S. Yu, B. Krishnapuram, R. Rosales, and R.B. Rao. Bayesian co-training. The Journal of Machine Learning Research, 999888:26492680, 2011.

[29] Dasgupta, S., M. L. Littman, and D. A. McAllester: 2001, PAC Generalization Bounds for Co-training.. In: NIPS. pp. 375382.

[30] Balcan, M.-F., Blum, A., and Yang, K. Co-training and expansion: Towards bridging theory and practice.In NIPS 17, pp. 8996. 2005.

[31] W.Wang and Z.H. Zhou. Analyzing co-training style algorithms. Machine Learning: ECML2007, pages 454465, 2007.

[32] Steven Abney. Bootstrapping. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pages 360367, 2002.

[33] Xu, X. 2003. Statistical Learning in Multiple Instance Problems. Master's thesis, University of Waikato.