Co-Labeling: A New Multi-view Learning Approach for Ambiguous Problems

Wen Li[1] Lixin Duan[2] Ivor Wai-Hung Tsang[1] Dong Xu[1]
[1] School of Computer Engineering, Nanyang Technological University, Singapore, 639798
[2] SAP Research, SAP AG, Singapore, 117440
wli1@e.ntu.edu.sg lxduan@gmail.com ivortsang@ntu.edu.sg dongxu@ntu.edu.sg

***Abstract*—We propose a multi-view learning approach called *co-labeling* which is applicable for several machine learning problems where the labels of training samples are uncertain, including semi-supervised learning (SSL), multi-instance learning (MIL) and max-margin clustering (MMC). Particularly, we first unify those problems into a general ambiguous problem in which we simultaneously learn a robust classifier as well as find the optimal training labels from a finite label candidate set. To effectively utilize multiple views of data, we then develop our co-labeling approach for the general multi-view ambiguous problem. In our work, classifiers trained on different views can teach each other by iteratively passing the predictions of training samples from one classifier to the others. The predictions from one classifier are considered as label candidates for the other classifiers. To train a classifier with a label candidate set for each view, we adopt the Multiple Kernel Learning (MKL) technique by constructing the base kernel through associating the input kernel calculated from input features with one label candidate. Compared with the traditional co-training method which was specifically designed for SSL, the advantages of our co-labeling are two-fold: 1) it can be applied to other ambiguous problems such as MIL and MMC; 2) it is more robust by using the MKL method to integrate multiple labeling candidates obtained from different iterations and biases. Promising results on several real-world multi-view data sets clearly demonstrate the effectiveness of our proposed co-labeling for both MIL and SSL.**

***Keywords*-ambiguous learning; multi-instance learning; semi-supervised learning; multiple kernel learning; TBIR**

I. Introduction

In many real world applications, it is generally difficult to obtain a sufficient number of labeled samples to learn robust classifiers. Therefore, researchers have been exploiting various learning scenarios by learning from ***ambiguous data*** whose labels are uncertain but under certain constraints. Taking document classification as an example, it is usually convenient to collect a large number of unlabeled documents; however, it is too costly to annotate all of them with correct labels. A practical way is to train the classifier by using only a small number of labeled documents together with a large number of unlabeled documents, which is also known as semi-supervised learning (SSL). Usually, the unlabeled samples in SSL are associated with a balance constraint to avoid biased solutions. Another example is the text-based web image retrieval (TBIR) [16]. Given a textual query, relevant images are retrieved based on the noisy textual description associated with each image. However, not all relevant images are truly positive with respect to the semantic meaning of the textual query. Li et al. [16] grouped relevant images into bags such that it is of high probability that each bag contains some truly positive images. Based on that, the TBIR task was modeled as a Multi-Instance Learning (MIL) problem, in which only the label of each bag is known and the labels of images in each bag remain unknown. In the literature, other learning scenarios (such as max-margin clustering (MMC) [26] and semi-supervised multi-instance learning (SSMIL) [18]) have also been widely studied to exploit ambiguous data. However, these existing methods were specifically designed for a certain learning scenario. In this paper, we generalize those learning scenarios as a unified ***ambiguous problem*** to utilize the various kinds of ambiguous data.

When multiple views of features are available, the information from different views can be effectively utilized to improve the performance for the learning task. As one pioneering work on multi-view learning, co-training [4] was proposed to solve the two-view SSL problem by simultaneously learning two classifiers on the two-view training data. Those two classifiers teach each other by iteratively annotating a certain amount of unlabeled data and putting them into the labeled training set. However, the co-training was specifically designed for SSL. It is difficult to apply to other learning problems such as MIL, since the sample selection process may violate the bag constraints in MIL (see Section III-A for a detailed discussion on co-training). Other work on multi-view SSL or multi-view clustering can also be found in [21], [15], [13], [14]. While those multi-view learning methods have shown advantages of utilizing multi-view information, they are all limited to a particular learning scenario and cannot be used to solve the general ambiguous problem with multi-view training data.

In this work, we address the general ambiguous problem from a multi-view perspective. First, the ambiguous data contain some samples with uncertain labels that satisfy some constraints (for example the bag constraints in MIL and the balance constraint in SSL), which means there are many possible labelings for the training samples. Therefore, the ambiguous learning problem can be treated as a task of learning from the training samples associated with a labeling set which contains many possible labelings. We call each labeling as a ***label candidate*** and the labeling set as the

label candidate set. It can be verified that the traditional SSL, MIL and MMC are special cases of our ambiguous learning problem with different constraints on the labeling. A comparison between the traditional supervised learning and the ambiguous learning is illustrated in Figure 1.

In the multi-view scenario, we can also model the learning problem of each view as an ambiguous problem. To have different views teach and learn from each other, we propose a ***"co-labeling"*** approach to solve the multi-view ambiguous problem, in which the classifier on one view can help the classifiers on the other views by sharing the labeling of training samples. Specifically, we first train a classifier on each view and obtain the predictions of training samples using those classifiers. Then, those predictions are projected into the feasible label candidate set to guarantee that the constraints are satisfied. After that, the label candidate set of each view is updated by merging the label candidates from the predictions of the classifiers on the other views. We also generate different predictions by varying the bias to enhance the robustness. This process is repeated until the stop criterion is reached. Compared with the co-training, in which different views select and label the training samples to help each other, the advantages of our co-labeling are twofold: 1) By sharing label candidates instead of selecting samples, our co-labeling can be applied to the general ambiguous problem with various constraints on the training labels. 2) By using the label candidates from different iterations and biases, our co-labeling can cope with possible mistakes in labeling and enhance robustness.

To learn from a label candidate set, we formulate a max-margin based model and then relax it to an MKL problem, which can be easily solved by existing MKL solvers [19], [27]. We also give the convergence analysis and the time complexity analysis. Finally, we conduct extensive experiments for multi-view SSL and multi-view MIL as well as some detailed experiential analysis.

Our main contributions are summarized as follows:

- From the perspective of label candidates, we formulate a general single-view ambiguous learning problem which unifies the traditional SSL, MIL, MISSL and clustering into one formulation.
- With the ambiguous learning formulation, we propose a new multi-view approach called co-labeling to solve the general multi-view ambiguous problem. To our knowledge, this is the first work to study the general multi-view ambiguous problem. An MKL solution is also developed to instantiate the proposed co-labeling approach.
- Taking the examples of SSL with its application on webpage/document classification and MIL with its application on text-based image retrieval, we demonstrate the effectiveness of our proposed co-labeling and also present extensive experimental analysis.

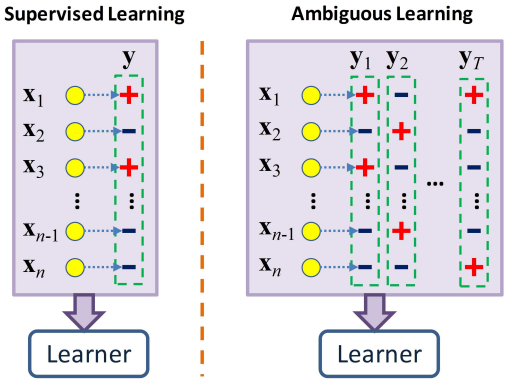


Figure 1. Illustration of the difference between supervised learning and ambiguous learning. Yellow circles denote the training samples, red crosses denote a positive label and blue strips denote a negative label. The green dashed rectangles that include red crosses and blue strips denote the labeling for training samples. **Left:** The labeling (green dashed rectangles) of training samples (yellow circles) in supervised learning is fixed. **Right:** In ambiguous learning, the labelings of training samples are uncertain, so the task is to learn a robust classifier from the training samples ($\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$) and a set of label candidates ($\{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$).

II. Ambiguous Learning from Label Candidates Perspective

In ambiguous learning, the training samples have uncertain labels that satisfy some constraints (for example, the bag constraints in MIL and the balance constraint in SSL), and the learning task is to learn a classifier based on those training samples as well as a label candidate set. In this work, we focus on the binary classification problem. Formally, let $\mathbf{X} = \{\mathbf{x}_i | i = 1, \ldots, n\}$ be the set of training samples where n is the total number of training samples. We use $\mathbf{y} = [y_1, \ldots, y_n]'$ to represent a possible labeling (label candidate) where $y_i \in \{+1, -1\}$ is the label of the i-th training sample. Then the label candidate set can be represented as $\mathcal{Y} = \{\mathbf{y}_t | t = 1, \ldots, T\}$ where $\mathbf{y}_t$ is the t-th label candidate and $T = |\mathcal{Y}|$ is the total number of label candidates. We also use $\mathbf{g} = [g_1, \ldots, g_n]'$ to represent the unseen ground-truth label vector of the training samples where $g_i \in \{+1, -1\}$ is the ground truth label of the i-th training sample.

A comparison of supervised learning and ambiguous learning is given in Figure 1. While in supervised learning the labeling of training samples $\mathbf{y}$ is known (see the left part of Figure 1), in ambiguous learning there are many possibilities on the labeling of training samples and the learning task is to learn a robust classifier from a set of label candidates (see the right part of Figure 1)). Based on the regularized empirical risk minimization principle [22], we formulate the ambiguous learning task as follows:

$$\min_{f, \mathbf{y} \in \mathcal{Y}} \quad \|f\|^2 + C \sum_{i=1}^{n} \ell(f, \mathbf{x}_i, y_i), \tag{1}$$

where $\mathcal{Y}$ is the label candidate set, f is the target classifier, and $\ell(\cdot)$ is the loss function.

The major difference between the ambiguous learning and the traditional supervised learning is that we need to infer the underlying labels vector $\mathbf{y}$ for training samples while learning the classifier. This is a non-trivial task since the cardinality of the label candidate set (*i.e.* T) is of a size exponential in terms of n. However, in Section III, we develop a simple but effective way to construct a small label candidate set in the multi-view scenario instead of using all possible label candidates. Then we give the detailed form of f and $\ell(\cdot)$ and formulate it as an MKL problem in Section IV. Here we provide some examples to show that several existing learning scenarios are actually special cases of our ambiguous learning.

Multi-Instance Learning: In Multi-Instance Learning, the training samples (instances) are organized into different sets (bags). The label of each training bag is known but the labels for the instances in the bag are unobserved. Usually the constraints on training samples are that all instances in the negative bags are negative and at least one instance in each positive bag is positive. Let us use $\mathcal{B}_I$ to denote the I-th training bag and Y_I to denote the corresponding bag label, then the constraints on the training samples can be represented as $\sum_{x_i \in \mathcal{B}_I}(y_i + 1)/2 \geq 1 \,\text{if}\, Y_I = 1$ and $y_i = -1$ otherwise [1]. In [16], more general constraints on positive bags were proposed by requiring at least a portion of each positive bag to contain positive instances. An iterative approach was used in [1] to infer the labeling $\mathbf{y}$ and an MKL formulation was used in [16] to learn the classifier by optimizing the labeling in the label candidate set.

Semi-supervised Learning: In semi-supervised learning, the training data include a limited number of labeled samples and a large number of unlabeled samples. Usually, the unlabeled samples are required to satisfy a balance constraint. It can also be formulated into our ambiguous learning formulation. Formally, suppose there are l labeled training samples and $n - l$ unlabeled training samples, then the constraints can be represented as $y_i = g_i$ for $i = 1, \ldots, l$ and $\sum_{i=l+1}^{n} y_i = \sigma$ where σ is a predefined parameter for the balance constraint. Existing semi-supervised learning algorithms cannot directly solve this formulation because the number of label candidates is too large, but we will show in the multi-view scenario how to learn with a small set of label candidates in the next section.

III. Multi-view Ambiguous Learning from Label Candidates Perspective

In the multi-view learning, each training sample is represented with different views of features. Formally, the training sample $\mathbf{x}_i$ is in the form of $(\mathbf{x}_i^1, \ldots, \mathbf{x}_i^V)$ where $\mathbf{x}_i^v$ is the feature in the v-th view and $v = 1, \ldots, V$. Typically, a classifier f^v is trained on the v-th view and the final classifier is fused by using the classifiers from all views, *i.e.* $f(\mathbf{x}) = \frac{1}{V}\sum_{v=1}^{V} f^v(\mathbf{x}^v)$.

We have formulated the single-view ambiguous learning as a learning problem that simultaneously optimizes the classifier f and the unknown training labels $\mathbf{y}$. In this section, we show that the multi-view learning can also be treated as a problem of learning from the label candidate set, and then we extend the single-view ambiguous learning to the multi-view scenario.

A. Feed Samples: A Review of Co-training

One of the pioneering works on multi-view learning is the co-training approach [4]. It was designed for semi-supervised learning problems with two views. Let us assume the labeled and unlabeled training sets are L_0 and U_0. Two classifiers f_0^1 and f_0^2 are trained on two views respectively using the initial labeled training set L_0. Then two sets of unlabeled training samples $\tilde{L}_0^1$ and $\tilde{L}_0^2$ are selected from the unlabeled set according to the prediction confidence, and are labeled as positive or negative by the two classifiers respectively. After that, the labeled training set is enlarged by merging the newly labeled data, *i.e.* $L_1 = L_0 \bigcup \tilde{L}_0^1 \bigcup \tilde{L}_0^2$ and the two classifiers are retrained on the enlarged labeled set. Such processes are repeated until a fixed number of iterations is reached.

The co-training can be seen as a process of iteratively feeding newly labeled training samples to each view. The classifiers of two views can be improved if the following assumptions hold: each view is sufficient to train a low-error classifier and both views are conditionally independent. The first assumption guarantees that the newly labeled samples are accurately labeled with high confidence and the second one ensures that the samples selected by one view are helpful to the other view. However, those assumptions usually do not strictly hold for real-world data. Therefore, many theoretical works on co-training from different perspectives tried to relax those assumptions, such as weak dependence [2], α-expansion [3], large diversity [24]and label propagation [25].

Although co-training has been applied in many applications, there are two major limitations which limit it from broader applications. First, it was specifically designed for SSL and cannot be used when there are other constraints on the unlabeled data. For example, in MIL, the data are provided in the form of bags, and directly using the sample selection strategy on instances may violate the constraints on bags. Second, the labels of the selected unlabeled data are fixed once they are labeled. If the classifier f^v is not robust enough to make accurate predictions, the wrongly labeled data will be propagated to subsequent retraining processes, which may significantly degrade the robustness of retrained classifiers.

B. Feed Label Candidates: The Co-Labeling Approach

Based on the general ambiguous problem formulation in (1), we propose our new multi-view approach for general

ambiguous learning problems as:

$$\min_{f^v, \mathbf{y}^v \in \mathcal{Y}^v} \quad \sum_{v=1}^{V} \left(\|f^v\|^2 + C \sum_{i=1}^{n} \ell(f^v, \mathbf{x}_i^v, y_i^v) \right), \qquad (2)$$

where f^v is the classifier on the v-th view, and $\mathcal{Y}^v$ is the label candidate set which is constructed by utilizing the predictions from the other views.

The algorithm of our multi-view approach is depicted in Algorithm 1. Different from co-training which iteratively updates the labeled training set by feeding newly labeled training samples, we let one view to help another by feeding its predictions to update the label candidate sets of another view. Therefore, we refer to this approach as ***co-labeling***. How to update the candidate set of one view using the predictions from the other views (*i.e.* line 5 in Algorithm 1) is the key of our co-labeling approach. To better illuminate the updating strategy, we detail it step by step by presenting three strategies, each of which is an improved version of the previous one. Denoting the label candidate set of the v-th view at the t-th iteration as $\mathcal{Y}_t^v$, we give the first strategy as follows:

Strategy 1: $\mathcal{Y}_{t+1}^v = \bigcup_{p \neq v}^{V} o_t^p$ *where* o_t^p *is obtained by projecting the decision value from the* p*-th view (*i.e.*, z^p*) into the feasible set* $\mathcal{Y}$ *defined by the constraints on the ambiguous training samples.*

In other words, the label candidate set is constructed by using the latest prediction from the other views. In this way, we treat the label vector as a whole which allows us to easily enforce the training labels to satisfy any constraints such as the bag constraints in MIL and the balance constraint in SIL and MMC. Therefore, the co-labeling does not have the first limitation of the co-training.

However, the second limitation has not been addressed yet. The classifiers may also be degraded if the label candidates are not accurate at one iteration and do harm to the classifiers trained at the next iteration. To improve robustness, instead of only using the latest prediction, we propose to construct the label candidate set for each view using the predictions of the other views from *all previous iterations*, which is formally stated in the following strategy:

Strategy 2: $\mathcal{Y}_{t+1}^v = (\bigcup_{p \neq v}^{V} o_t^p) \bigcup \mathcal{Y}_t^v$ *where* o_t^p *is obtained in the same manner as in Strategy 1.*

With this strategy, the label candidate set of each view is augmented as the number of iterations increases. If the newly obtained label candidates are not accurate, it is still possible to learn a classifier with the label candidates obtained from previous iterations. In other words, when the label candidates obtained from different iterations are not consistent, rather than using the latest one, we leave them to the learner to make the choice.

However, what if the label candidates from different iterations are consistent but wrong? This possibly happens for ambiguous learning problems since the learned classifiers in ambiguous problems are easily biased. For example, in MIL, one common approach is to initialize all the instances in positive bags to positive training samples, which makes the classifier more likely to predict the negative samples to be positive. Besides this, in SSL, the limited number of labeled data also easily cause the classifier to be biased. To handle this bias problem, instead of only using one prediction with fixed bias, we use multiple predictions with different biases to generate the label candidates, as inspired by the recent work on domain adaptation [20]. In other words, at the t-th iteration, instead of only obtaining one new label candidate o_t^p from the p-th view (see Strategy 2), the v-th view can obtain a set of new label candidates based on predictions with different biases. In implementation, it simply involves adding different biases to the original decision values $\mathbf{z}^v$. We formally introduce the new strategy as follows:

Strategy 3: $\mathcal{Y}_{t+1}^v = (\bigcup_{p \neq v}^{V} \mathcal{O}_t^p) \bigcup \mathcal{Y}_t^v$ *where* $\mathcal{O}_t^p$ *is a set of label candidates obtained in the same manner as in Strategy 1 from the predictions with different biases.*

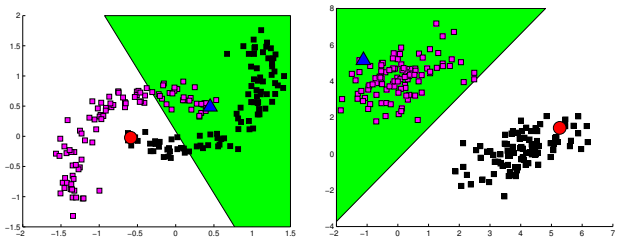
Algorithm 1 The Co-Labeling Algorithm

Require: Training samples with V views, initial labels $\mathbf{y}_0$.
Ensure: Classifier f^v's.
1: Initialize the label candidates set $\mathcal{Y}^v = \{\mathbf{y}_0\}$ for each view.
2: **repeat**
3: Train a classifier f^v based on $\mathcal{Y}^v$ on each view.
4: Get the prediction $\mathbf{z}^v$ of the training examples using f^v on each view.
5: Update the label candidate $\mathcal{Y}^v$ using $\mathbf{z}^p$'s for $p \neq v$.
6: **until** The stop criterion is reached.
7: **return** f^v's.

C. Discussion

We give a brief discussion of our co-labeling approach with respect to the co-training method in the SSL setting with a toy problem shown in Figure 2. View 1 contains two-moons data and View 2 contains two clusters. We have one positive sample and one negative sample denoted by the blue triangle and the red circle, respectively. The magenta and black rectangles denote the unlabeled samples in the two classes.

Suppose we use SVM as the base classifier for the co-training. In the first iteration, the co-training trains one classifier for each view using the labeled samples. We plot the decision boundaries of two views in Figure 2. It can be observed that the classifier trained on View 1 is not good since we only have two training samples. And next, in the sample selection step, the samples located at the tails of the two moons will be selected since they are farthest from the decision boundary. However, they are wrongly labeled and will degrade the classifiers trained in the next iteration.



(a) View 1 (b) View 2

Figure 2. The toy data with two views. Blue triangles denote labeled positive samples and red circles denote labeled negative samples. The magenta/black rectangles denote unlabeled samples in the two classes.

In contrast, our co-labeling does not suffer from this problem since the wrongly labeled samples will be corrected by other label candidates from different iterations and different biases. The final prediction results of two methods are shown in Section VI. Besides the robustness of our co-labeling approach, it also can be readily applied on other ambiguous learning scenarios such as MIL and MMC.

IV. The Implementation of Co-Labeling

We have proposed a uniform formulation of ambiguous learning and a new multi-view algorithm in the last section, both of which require learning a classifier (*i.e.* the f in Equation (1)) from a label candidate set. In this section, we give the detailed forms of f and $\ell(\cdot)$, and relax the problem to an MKL problem which can be solved easily with existing solvers.

A. The Formulation

We adopt the maximum margin classifier $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$ where $\phi(\cdot)$ is the feature mapping function, and use ρ-SVM with the squared hinge loss[1] to solve the ambiguous learning problem. Based on (2), we arrive at the following optimization problem for the v-th view:

$$\min_{\mathbf{y}^v \in \mathcal{Y}^v} \min_{\mathbf{w}^v, b^v, \rho^v, \xi_i} \quad \frac{1}{2}\left(\|\mathbf{w}^v\|^2 + {b^v}^2 + C\sum_{i=1}^{n}\xi_i^2\right) - \rho^v, \tag{3}$$

$$\text{s.t.} \quad y_i^v(\mathbf{w}^{v'}\phi(\mathbf{x}_i^v) + b^v) \geq \rho^v - \xi_i,\ i = 1, \ldots, n,$$

where C is the tradeoff parameter, $2\rho^v/\|\mathbf{w}^v\|$ defines the margin and y_i is the i-th element of the label vector $\mathbf{y}$. The superscript v denotes the v-th view, in other words, we shall optimize the above objective function on each view. For simplification, we omit the superscript v unless necessary.

B. The MKL solution

By introducing the dual variable $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]'$ for the constraints in (3), we derive the dual problem of (3) as:

$$\min_{\mathbf{y} \in \mathcal{Y}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} -\frac{1}{2}\boldsymbol{\alpha}'\left(\mathbf{K} \circ \mathbf{y}\mathbf{y}' + \frac{1}{C}\mathbf{I}\right)\boldsymbol{\alpha}, \tag{4}$$

[1] It has been suggested in [28] that the squared hinge loss is more robust than the hinge loss for solving the MMC problem.

where $\mathbf{K} = \hat{\mathbf{K}} + \mathbf{1}\mathbf{1}'$, $\hat{\mathbf{K}} = [k(\mathbf{x}_i, \mathbf{x}_j)] = [\phi(\mathbf{x}_i)'\phi(\mathbf{x}_j)]$ is the kernel matrix, and $\mathcal{A} = \{\boldsymbol{\alpha} | \boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\alpha}'\mathbf{1} = 1\}$ is the feasible set of $\boldsymbol{\alpha}$.

Instead of directly solving the mixed-integer problem in (4), we seek to optimize the linear combination of $\mathbf{y}\mathbf{y}'$'s. Then, the above problem is relaxed to an MKL problem which is a lower bound of (4). The relaxed formulation is as follows:

$$\min_{\mathbf{d} \in \mathcal{D}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \quad -\frac{1}{2}\boldsymbol{\alpha}'\left(\sum_{t=1}^{|\mathcal{Y}|} d_t \mathbf{K} \circ \mathbf{y}_t\mathbf{y}_t' + \frac{1}{C}\mathbf{I}\right)\boldsymbol{\alpha}, \tag{5}$$

where $\mathbf{d}$ is a vector of combination coefficients for the base kernels $\mathbf{K} \circ \mathbf{y}_t\mathbf{y}_t'$'s, and $\mathcal{D} = \{\mathbf{d} | \mathbf{d} \geq \mathbf{0}, \mathbf{d}'\mathbf{1} = 1\}$ is the feasible set of $\mathbf{d}$. For more details about the above relaxation, please refer to [17], [16]. Note that the $\mathcal{Y}$ is the label candidate set on each view, and its cardinality equals the number of iterations in Algorithm 1, which is very small. Therefore it can be efficiently solved by existing MKL solvers [27]. Although [17], [16] also formulated the MMC and MIL problem as an MKL problem in a similar way, they are single-view approaches and the motivations are totally different. Moreover, in their methods, the label candidates were obtained by iteratively finding the most violated constraint, which is another NP-hard problem.

After solving the MKL on each view, the final classifier is given by:

$$f(\mathbf{x}) = \sum_{v=1}^{V} f^v(\mathbf{x}^v) = \sum_{v=1}^{V} \frac{1}{\rho_v}\left(\sum_{i=1}^{n}\alpha_i^v \sum_{t=1}^{|\mathcal{Y}^v|} d_t^v y_{t,i}^v (k(\mathbf{x}_i^v, \mathbf{x}^v) + 1)\right).$$

It is worth noting that the prediction is as fast as the SVM prediction on each view.

C. The Algorithm

We summarize our detailed algorithm in Algorithm 2. The initial labels $\mathbf{y}_0$ are problem dependent. For example, in SSL we can use the prediction of the classifier trained on the labeled data and in MIL we usually initialize all instances in positive bags as positive and all instances in negative bags as negative. After the initialization, the classifier will be trained on each view by solving (5). Then we use the classifier on each view to predict the labels of training samples, and obtain a set of predictions by varying the bias. All the predictions are required to satisfy the constraints (*i.e*, bag constraints in MIL or balance constraint in SSL). Finally the label candidate set of each view is augmented by merging the predictions from the other views. This process is repeated until the stop criterion is reached.

The Constraints: We use a simple method to force the prediction to satisfy the constraints on ambiguous data. Two typical constraints are discussed in Section II which are commonly used in SSL and MIL. We first sort the decision values of all training samples in descending order. The labels for labeled samples are assigned using their ground-truth

Algorithm 2 The Algorithm of Co-Labeling with MKL

Require: Training samples with V views, initial labels $\mathbf{y}_0$.
Ensure: Classifier variables $\mathbf{d}^v$'s, $\boldsymbol{\alpha}^v$'s and label candidate sets $\mathcal{Y}^v$'s.

```
Initialize the label candidates set Y^v = {y_0} for each view.
repeat
    Solve α^v and d^v in (5) based on Y^v.
    Get the decision values of training data z^v using α^v and d^v.
    Vary the biases to obtain a set of decision values Z^v for each view.
    Get the label candidate set O^v by enforce each prediction in Z^v to satisfy the constraints.
    Set each Y^v = Y^v ∪ O^p for any p ≠ v.
until The stop criterion is reached.
return α^v's, d^v's and Y^v's.
```

training labels. For the remaining training samples, in SSL we simply label the first m samples as positive and the remaining as negative where m is determined by the balance constraint, and in MIL we label the first m_I instances in the I-th positive bag as positive and the remaining as the sign of their decision values where m_I is determined by the constraint on the I-th positive bag. It can be verified that in this way the constraints discussed in Section II can be satisfied.

Stop Criterion: One observation is that the objective value in Algorithm 2 decreases monotonously on each view. The reason is as follows: We solve an MKL problem which minimizes the objective function in (5) with respect to $\boldsymbol{\alpha}^v$ and $\mathbf{d}^v$. Since the label candidate set $\mathcal{Y}^v$ is augmented at each iteration, in the worst case the optimal solution of MKL at the current iteration should be the same as that at the last iteration by setting the entries in the coefficient vector $\mathbf{d}^v$ corresponding to the newly added label candidates to zeros. Therefore the objective values of MKL on each view should decrease monotonicly as the number of iterations increases. Experimental results of convergence are presented in Section VI-B4. So the stop criterion in the co-labeling is that the difference of the objective values between two iterations is less than a small value on all views or the maximum number of iterations is reached. Usually, we observe thast the algorithm runs fast and stops in fewer than 10 iterations.

Time Complexity: The main cost in Algorithm 2 is the training process of MKL and the testing process on the training samples. Since the kernel was computed in the training process, the testing process is only a matrix multiplication operator and the cost can be ignored compared with the training process. Considering the non-linear case, let us denote the complexity of training an MKL as $O(\text{MKL})$ [2] and suppose our algorithm runs T iterations, then the total time complexity of Algorithm 2 is $T \cdot V \cdot O(\text{MKL})$ where V is the total number of views. For the linear case, there exists a fast algorithm to solve the SVM problem in the primal form, so our co-labeling can be much faster than in the non-linear case.

V. Related Work

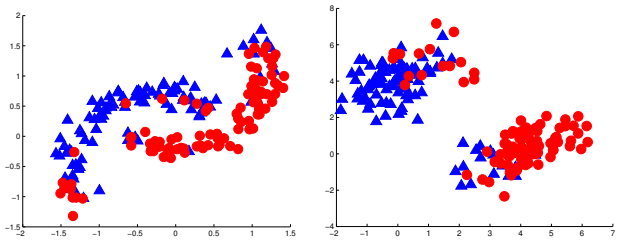
Our work is related to multi-view learning. As the pioneering work on multi-view learning, Blum and Mitchell [4] introduced the co-training approach for semi-supervised learning. The original assumption of co-training that two views are conditionally independent is too strong for real applications. Therefore, different explanations were proposed to analyze co-training under more relaxed assumptions such as weak dependence [2], α-expansion [3], large diversity [24] and label propagation [25]. In [21], a co-regularization approach was proposed to minimize the disagreement of the classifiers of two views. Christoudias et al. [8], [9] studied the co-training problem with noisy observations and Li et al. [15] extended the transductive SVM with co-regularization. Recently, Chen et al. [7] proposed a feature decomposition approach for co-training when only one single view exists and recent work [13], [14] extended the co-training and co-regularization to multi-view spectral clustering. However, those works are restricted in the multi-view semi-supervised setting or were specifically designed for certain cases. In contrast, our co-labeling approach is a general learning framework for multi-view learning on any data with ambiguous labels and different constraints.

Our work is also related to the various cases of ambiguous learning. More work on semi-supervised learning was summarized in [30] and on multiple instance learning in [29]. For max-margin clustering, readers can refer to [26], [28] and [17]. The most related work are LGMMC [17] and MIL-CPB [16]. The MKL algorithm was also used in the two papers to solve the learning problem. However, our proposed approach is intrinsically different with these two works. Our co-labeling approach is motivated from co-training and aims to study how to effectively utilize the information from different views which is applicable to different general ambiguous scenarios. In contrast, those works were designed for a certain case of ambiguous learning (MMC or MIL) in the single-view setting.

VI. Experiments

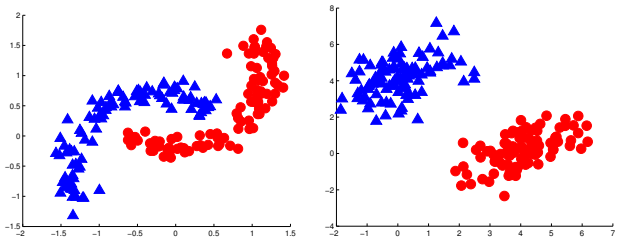
In this section, we first compare our co-labeling with the traditional co-training on the toy problem mentioned in Section III-C. Then we evaluate our co-labeling approach

[2]The time complexity of MKL has not been theoretically analyzed. Usually, the MKL solver needs to train an SVM for tens of iterations. The empirical analysis shows that optimizing the QP problem in SVM is $O(n^{2.3})$ where n is the number of training samples. Therefore, the complexity of MKL is $O(kn^{2.3})$ where k is the number of iterations in MKL.



(a) View 1 (b) View 2

Figure 3. The prediction on the toy data by co-training. The blue triangles denote the predicted positive samples and red circles the predicted negative samples.



(a) View 1 (b) View 2

Figure 4. The prediction on the toy data by our co-labeling. The blue triangles denote the predicted positive samples and red circles the predicted negative samples.

on several data sets for two cases: 1) Multi-View Semi-Supervised Learning (MVSSL), 2) Multi-View Multiple Instance Learning (MVMIL) .

A. The Toy Problem

The training data of the toy problem are shown in Figure 2. View 1 contains the two-moons data[3] and View 2 contains two clusters which are randomly generated with the same covariance matrix $\sigma = [1\ 0.5; 0.5\ 1]$ and with the centers $[0; 4]$ and $[4; 0]$, respectively.

We use SVM as the base classifier to implement the co-training algorithm. Five positive samples and five negative samples are selected at each iteration. The classifiers trained on two views are fused by using the average decision value. The final predictions of the two classifiers are shown in Figure 3 and Figure 4, respectively. It can be seen in Figure 3 that the wrongly labeled data in the first iteration (the samples located at the tails of the two moons) degrade the final classifier. The final predictions on those samples are still not correct. Although our co-labeling also makes wrong predictions on those samples in the first iteration since we also train the classifier to initialize the labeling for unlabeled data in the SSL setting (see Section IV), it correctly predicts all the samples using the final classifier because our co-labeling can correct the errors by leveraging the label candidates obtained from different iterations and different biases. The prediction accuracies of the co-training and our co-labeling are 83.00% and 100.00% respectively.

[3]It is downloaded from http://www.dii.unisi.it/ melacci/lapsvmp/

B. Multi-View Semi-Supervised Learning

1) Data sets: We evaluate our co-labeling approach for semi-supervised learning on two applications, news classification and webpage classification. The BBC data set and BBCSport data set are used for news classification; and the WebKB data set is used for webpage classification. The details of these data sets are summarized in Table I and described as follows:

Table I
SUMMARIZATION OF THE DATA SETS USED IN SSL. D1 AND D2 ARE THE FEATURE DIMENSIONS OF TWO VIEWS. $\#c$, $\#l$, $\#u$ AND $\#t$ ARE THE NUMBERS OF CLASSES, LABELED DATA, UNLABELED DATA AND TEST DATA, RESPECTIVELY.

Data sets	d1	d2	$\#c$	$\#l$	$\#u$	$\#t$
BBC	4817	4818	5	10	1104	1111
BBCSport	2306	2307	5	10	360	367
WebKB	3000	1840	2	12	1039	

The BBC and BBCSport data sets: The two data sets [12] contain news articles collected from the BBC[4]. The BBC data set includes 2225 documents from five topics (business, entertainment, politics, sports and technology) and the BBCSport data set includes 737 sports news documents from five classes (athletics, cricket, football, rugby and tennis). We randomly segment the feature into two views as in [13]. We partition the data sets into the training set and the test set, each of which contains 50% of the documents per class. Ten labeled data are further selected from the training set.

The WebKB data set: The WebKB data set has been widely used to evaluate multi-view methods [4], [21], [15]. It contains 1051 web pages belonging to two categories: course or non-course. For each web page, there are two views, the page view which contains the textual content of this page, and the link view which contains the anchor-text on links from other webpages pointing to this page. 3000-d features are extracted for the page view and 1840-d features for the link view, respectively. 12 samples are selected as the labeled data and the remaining samples are used as unlabeled data[5].

2) Experimental Setting: We compare our co-labeling with following baselines:

- SVM: The standard SVM trained with the labeled data, which is a commonly used baseline in semi-supervised learning.
- TSVM [10]: The transductive SVM trained with the labeled and unlabeled data. The two views are later fused to obtain the final result.

[4]The features can be downloaded from http://mlg.ucd.ie/datasets/bbc.html

[5]The indices of labeled data and features are publicly available in http://people.cs.uchicago.edu/∼vikass/manifoldregularization.html

Table II
PERFORMANCES OF DIFFERENT METHODS ON THREE DATA SETS. THE BEST RESULTS ARE DENOTED IN BOLDFACE. MAPS AND STANDARD DEVIATIONS ARE REPORTED ON BBC&BBCSPORT. ↑ DENOTES THE RESULT IS SIGNIFICANTLY BETTER THAN THE OTHERS ACCORDING TO THE T-TEST WITH A SIGNIFICANCE LEVEL AT 0.05. PRBEPS ARE REPORTED ON WEBKB. THE RESULTS OF SVM, TSVM AND CO-LAPSVM ARE FROM TABLE 1 IN [21]. SINCE THE STANDARD DEVIATIONS OF THESE METHODS ARE NOT AVAILABLE, WE ONLY REPORT THE AVERAGE PRBEPS ON WEBKB.

	BBC			BBCSport			WebKB		
	View1	View2	View1+2	View1	View2	View1+2	page	link	page+link
SVM	66.53(4.08)	63.11(3.67)	74.26(3.37)	70.69(3.42)	66.43(3.98)	76.99(3.76)	74.4	77.8	84.4
TSVM	71.99(5.48)	66.83(3.54)	75.72(3.16)	74.62(5.73)	65.51(3.36)	79.21(6.43)	85.5	91.4	92.2
Co-LapSVM	70.30(3.39)	68.04(4.56)	76.97(3.41)	70.70(3.43)	66.43(3.99)	77.14(3.29)	**94.3**	**93.3**	94.2
2V-TSVM	52.70(3.96)	52.61(5.34)	58.39(5.25)	64.00(3.08)	63.50(3.86)	69.82(3.78)	85.7	86.7	87.3
PMC	—	—	71.57(6.37)	—	—	79.48(5.41)	—	—	88.6
Co-Labeling	**78.41(3.79)**↑	**77.61(3.01)** ↑	**81.37(3.14)** ↑	**82.10(5.41)** ↑	**79.60(4.44)** ↑	**84.22(5.11)** ↑	92.5	93.1	**95.1**

- Co-LapSVM [21]: The Laplacian SVM in the multi-view setting.
- 2V-TSVM [15]: The TSVM version of co-regularization, in which both ramp loss and co-regularization are used to cope with the two-view setting.
- PMC [7]: An improved version of co-training which is designed to split the single view features into two views. We apply PMC on the joint-view of these three data sets. Due to the randomness in initializing the feature splits, we run each split five times and use the split with the minimum objective value of the feature split algorithm PMD (see [7]).

A binary classifier is trained for all the methods, and linear kernels are used. The experiments on the BBC and BBCSport data sets are repeated on 10 training/testing data splits. The experiments on the WebKB data set are run with 100 training/testing data splits as in [21].

3) Performance: The results on these three data sets are reported in Table II. Mean average precisions (MAPs) of 5 classes over 10 rounds are reported for the BBC and BBCSport data sets. Following [21], the average of precision-recall break-even point (PRBEP) over 100 splits are reported for the WebKB data set. The results of SVM, TSVM and Co-LapSVM on the WebKB data set are copied from Table 1 in [21].

From the results, we can see that our method achieves the best results for the joint-view, which demonstrates the advantage of our co-labeling for semi-supervised learning. The improvements over the second best in the three data sets are 4.40%, 4.74% and 0.9%, respectively. It is worth noting that the co-labeling is significantly better than other methods on the BBC and BBCSport data sets for each single view as well as the joint-view. Although the co-labeling does not achieve best results for the page view and link view on the WebKB data set, the late fusion result is the best which again demonstrates that our method can effectively use multi-view information. The relative improvement is not as significant as on the BBC and BBCSport data sets because it already has a very high performance, which can be confirmed by the fact our method achieves 99.11% in the measurement of MAP over 100 splits.

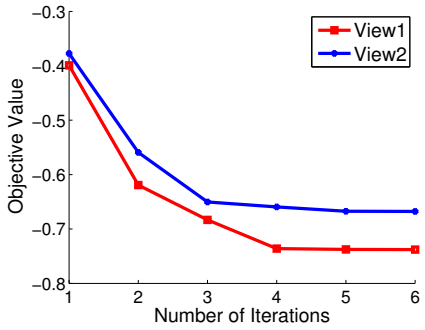


Figure 5. The objective values with respect to the number of iterations on two views.

Table III
PERFORMANCE OF DIFFERENT VERSIONS OF CO-LABELING ON BBC-SPORTS.

	View1	View2	View1+View2
Co-Labeling(iter)	58.19	61.56	69.77
Co-Labeling(nobias)	74.21	72.00	77.43
Co-Labeling	82.10	79.60	84.22

4) Convergence Analysis: We have analyzed that the objective value of MKL on each view decreases with respect to the number of iterations (see Section IV-C). Taking the first topic (*i.e.* "athletics") of the BBC-Sports data set as an example, we show the objective values of two views in Figure 5. It can be observed the objective values of the two views converge very fast and it takes only six iterations to reach the stop criterion. The average number of iterations for all 5 classes over 10 splits is 6.36.

5) Strategy Analysis: Three strategies for updating the label candidate set are discussed in Section III-B. Here we evaluate those three strategies by introducing two simplified versions of our co-labeling algorithms: the first version is *Co-Labeling(iter)*, in which we only use the labeling from the latest iteration (Strategy 1); and the other one, *Co-Labeling(nobias)*, uses labelings from all previous iterations but does not vary biases at each iteration (Strategy 2). We also take the BBCSport data set as an example, and the results of those two simplified versions of our co-labeling approach are shown in Table III. We can observe that the performance of Co-Labeling(iter) is degraded without using

Table IV
AVERAGE TRAINING TIMES OF DIFFERENT MULTI-VIEW SSL METHODS PER CLASS PER SPLIT ON THREE DATA SETS (IN SECONDS).

	BBC	BBCSport	WebKB
Co-LapSVM	52.45	2.136	16.69
2V-TSVM	1108	497.3	446.4
PMC	30.64	7.215	55.41
Co-Labeling	36.50	5.111	21.27

MKL and the bias. It is even worse than the results of SVM reported in Table II which only uses the labeled data. This is not surprising since the labeling from the trained classifiers may be very noisy. Directly feeding the labeling as the training labels to the other view cannot guarantee a robust classifier as discussed in Section III-B. After using MKL, the performances of Co-Labeling(nobias) are improved a lot, which are comparable to other multi-view methods as reported in Table II. However, Co-Labeling(nobias) still suffers from the bias problem since we use the prediction as labeling while other methods do not have this issue. After varying the biases (Strategy 3), the final results are significantly better than other methods.

6) Time Comparison: The training time of different multi-view methods are reported in Table IV. All methods are performed on a workstation with a 3.3GHz CPU. We implement the co-labeling with unoptimized MATLAB code using LIBSVM[6]. We observe that the co-labeling is comparable to the Co-LapSVM and PMC in training time on the three data sets. It demonstrates that the co-labeling is practical and scalable for real applications considering it can be further sped up after a better implementation and using a faster solver (for example LIBLINEAR for the linear case).

C. Multi-View Multiple Instance Learning

MIL has been successfully exploited in Text-Based Image Retreival (TBIR), so we also evaluate our co-labeling approach for TBIR under the multiple view setting. Similar to [16], we conduct the experiment on the large-scale NUS-WIDE data set, which consists of 269,648 images from 81 annotated concepts collected from the website *Flickr.com*. Three groups of features are extracted:

- 1) **The textual feature** is extracted from the associated tags for each image. The vocabulary is constructed using 1000 words with the highest frequency. Then, a 1000 dimensional term-frequency feature is extracted for each image.
- 2) **The global visual features** are extracted according to the procedure in [16], then we concatenate three types of global features and apply PCA to obtain 119-d features.
- 3) **The local visual feature** is based on the SIFT feature by preserving LCC coding, which is extracted for each image using the method in [23]. We train a codebook with 4096 visual words and use a three-level spatial pyramid, and finally we get an 86016 dimensional sparse vector for each image.

[6]It can be further accelerated by using LIBLINEAR which solves the SVM in the primal form.

Table V
MEAN AVERAGE PRECISIONS (MAPS) OVER 81 CONCEPTS FROM DIFFERENT METHODS ON THE NUS-WIDE DATA SET.

	TG	TL	TG+TL
MIL-CPB	61.43	57.84	77.07
mi-SVM	59.25	59.26	77.18
sMIL	60.01	62.09	75.48
Co-Labeling	**62.56**	61.71	**79.09**

Two views are constructed by using these three types of features. Specifically, we partition the 1000-d textual feature into two 500-d vectors, $\mathbf{t}_1$ and $\mathbf{t}_2$, with even dimensions and odd dimensions respectively. Then we further concatenate the two textual vectors with the global visual feature vector and local visual feature vector to form the final feature vectors for two views with the form $\mathbf{x} = [\gamma\mathbf{t}; \lambda\mathbf{v}]$, which are denote as TG view and TL view, respectively. For the TG view, following [16], we set $\gamma = 1$ and $\lambda = 0.1$. For TL view, we empirically set $\gamma = 0.1$ and $\lambda = 0.9$.

We compare our co-labeling approach with three MIL methods, MIL-CPB [16], mi-SVM [1] and sMIL [6] which have the best performances as studied in [16]. Since they are single-view methods, we use the late fusion method to average the decision values of the classifiers from different views [7]. For all methods, we construct 25 positive bags using the top relevant images and 25 negative bags using randomly selected irrelevant images, with each bag containing 15 instances. A Gaussian kernel is used on the TG view and a linear kernel is used on the TL view. A binary classifier is trained for each concept, and the top-100 Mean Average Precisions (MAPs) are reported in the experiments.

The MAPs over 81 concepts for different methods on two views as well as the results after using late fusion on the NUS-WIDE data set are reported in Table V. We have the following observations:

- Our co-labeling approach achieves the best late fusion result, which demonstrates the effectiveness of the proposed method. The improvement over the second best is 1.9% in terms of MAP over 81 concepts.
- The performances of baseline methods on the TG view are consistent with the reported results in [16] and our method also achieves the best result.
- On the TL view, we observe that the instance-based methods MIL-CPB and mi-SVM are worse than the bag-based method sMIL. A possible explanation is the labels inferred in the learning process are quite noisy because the data cannot be well separated by using a linear classifier. However, the co-labeling still obtains a good result as it considers not only the input TL features but also uses the outputs from the TG view.

Moreover, the late fusion result of sMIL is worse than other methods, although it has good performance on each view.

[7]We also tried using early fusion (the average kernel) for these methods. However, the results are worse than those of late fusion.

It is more likely the classifiers from two views are not as complementary to each other as in other methods. A possible reason is that the bag-level MIL methods cannot effectively use all information on training instances [16]. It also demonstrates the advantage of multi-view methods which can effectively use the information from different views.

VII. Conclusions and Future Work

To effectively utilize different types of multi-view ambiguous data, in this paper we have formulated a unified multi-view framework which covers various ambiguous learning problems including SSL and MIL under the multi-view setting. We firstly propose a general method to solve the single-view ambiguous learning problem using label candidates. Then, a unified framework is proposed for multi-view ambiguous learning tasks in which the label candidate sets are constructed by using the label predictions from the classifiers trained on the other views. MKL is used to train a robust classifier from a set of label candidates for each view. Extensive experimental results on both MIL and SSL with real-world data sets demonstrate the effectiveness of our proposed approach.

In the future, we plan to investigate the proposed approach on other ambiguous learning tasks, such as MISSL and MMC.

Acknowledgment

This research is supported by the National Research Foundation Singapore under its Interactive & Digital Media (IDM) Public Sector R&D Funding Initiative (Grant No. NRF2008IDM-IDM004-018) and administered by the IDM Programme Office.

References

[1] Andrews, S., Tsochantaridis, I., and Hofmann, T. Support vector machines for multiple instance learning. In *NIPS*, 2003.

[2] Abney, S. Bootstrapping. In *ACL*, pp. 360-367, 2002.

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

[4] Blum, A. and Mitchell, T. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

[5] Brefeld, U., Gärtner, T., Scheffer, T., and Wrobel, S. Efficient co-regularised least squares regression. In *ICML*, 2006.

[6] Bunescu, Razvan C. and Mooney, Raymond J. Multiple instance learning for sparse positive bags. In *ICML*, 2007.

[7] Chen, M., Weinberger, K., and Chen, Y. Automatic feature decomposition for single view co-training. In *ICML*, 2011.

[8] Christoudias, C. M., Urtasun, R., and Darrell, T. Multi-view learning in the presence of view disagreement. In *COLT*, 2008.

[9] Christoudias, C. M., Urtasun, R., Kapoor, A., and Darrell, T. Co-training with noisy perceptual observations. In *CVPR*, 2009.

[10] Collobert, R., Sinz, F., Weston, J., and Bottou, L. Large scale transductive svms. *JMLR*, 7:1687–1712, 2006.

[11] Dasgupta, S., Littman, M. L., and McAllester, D. Pac generalization bounds for co-training. In *NIPS*, 2001.

[12] Greene, D. and Cunningham, P. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *ICML*, 2006.

[13] Kumar, A. and Daumé III, H. A co-training approach for multiview spectral clustering. In *ICML*, 2011.

[14] Kumar, A., Rai, P., and Daumé III, H. Co-regularized multi-view spectral clustering. In *NIPS*, 2011.

[15] Li, G., Hoi, S. C.-H., and Chang, K. Two-view transductive support vector machines. In *SDM*, 2010.

[16] Li, W., Duan, L., Xu, D., and Tsang, I. W. Text-based image retrieval using progressive multi-instance learning. In *ICCV*, 2011.

[17] Li, Y.-F., Tsang, I. W., Kwok, J.T., and Zhou, Z.-H. Tighter and convex maximum margin clustering. In *AISTATS*, 2009.

[18] Rahmani, R. and Goldman, S. A. MISSL: Multiple-instance semi-supervised learning. In *ICML*, 2006.

[19] Rakotomamonjy, A., Bach, F., Grandvalet, Y. and Canu, S. SimpleMKL. Journal of Machine Learning Research (JMLR), Vol. 9, pp 2491-2521, 2008

[20] Seah, C.-W., Tsang, I. W. and Ong, Y.-S. Healing Sample Selection Bias by Source Classifier Selection. In *ICDM*, 2011.

[21] Sindhwani, V., Niyogi, P., and Belkin, M. A co-regularization approach to semi-supervised learning with multiple views. In *ICML*, 2005.

[22] Vapnik, V. *Statistical learning theory*. Wiley-Interscience, 1998.

[23] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[24] Wang, W. and Zhou, Z.-H. Analyzing co-training style algorithms. In *ECML*, 2007.

[25] Wang, W. and Zhou, Z.-H. A new analysis of co-training. In *ICML*, 2010.

[26] Xu, L., Neufeld, J., Larson, B., and Schuurmans, D. Maximum margin clustering. In *NIPS*, 2005.

[27] Xu, Z., Jin, R., Yang, H., King, I., and Lyu, Michael R. Simple and efficient multiple kernel learning by group lasso. In *ICML*, 2010.

[28] Zhang, K., Tsang I. W. and Kwok, J.T. Maximum Margin Clustering Made Practical. IEEE Transactions on Neural Networks, 20(4): 583-596, 2009.

[29] Zhou, Z.-H. Multi-instance learning: A survey. Technical report, AI Lab, Department of Computer Science & Technology, Nanjing University, Nanjing, China, 2004.

[30] Zhu, X. Semi-supervised learning literature survey. Technical report, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, 2006.