Research Article

Online Coregularization for Multiview Semisupervised Learning

Boliang Sun, Guohui Li, Li Jia, and Kuihua Huang

College of Information System and Management, National University of Defense Technology, Changsha, Hunan 410073, China

Correspondence should be addressed to Boliang Sun; sunboliang@nudt.edu.cn

Received 27 June 2013; Accepted 5 August 2013

Academic Editors: G. Kou and K. C. Patidar

Copyright © 2013 Boliang Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a novel online coregularization framework for multiview semisupervised learning based on the notion of duality in constrained optimization. Using the weak duality theorem, we reduce the online coregularization to the task of increasing the dual function. We demonstrate that the existing online coregularization algorithms in previous work can be viewed as an approximation of our dual ascending process using gradient ascent. New algorithms are derived based on the idea of ascending the dual function more aggressively. For practical purpose, we also propose two sparse approximation approaches for kernel representation to reduce the computational complexity. Experiments show that our derived online coregularization algorithms achieve risk and accuracy comparable to offline algorithms while consuming less time and memory. Specially, our online coregularization algorithms are able to deal with concept drift and maintain a much smaller error rate. This paper paves a way to the design and analysis of online coregularization algorithms.

1. Introduction

Semi-supervised learning (S^2L) is a relatively new subfield of machine learning which has become a popular research topic throughout the last two decades [1–6]. Different from standard supervised learning (SL), the S^2L paradigm learns from both labeled and unlabeled examples. In this paper, we investigate the online semi-supervised learning (OS^2L) problems with multiple views which have four features: (1) data is abundant, but the resources to label them are limited; (2) data arrives in a stream and cannot store them all; (3) the target functions in each view agree on labels of most examples (compatibility assumption); (4) the views are independent given the labels (independence assumption).

OS^2L algorithms take place in a sequence of consecutive rounds. On each round, the learner is given a training example and is required to predict the label if the example is unlabeled. To label the examples, the learner uses a prediction mechanism which builds a mapping from the set of examples to the set of labels. The quality of an OS^2L algorithm is measured by the cumulative loss it makes along its run. The challenge of OS^2L is that we do not observe the true label for unlabeled examples to evaluate the performance of prediction mechanism. Thus, if we want to update the prediction mechanism, we have to rely on indirect forms of feedback.

Lots of OS^2L algorithms have been proposed in recent years (see a survey in [7, 8]). A popular idea is defining an instantaneous risk function and decreasing its value in an online manner to avoid optimizing the primal semi-supervised problem directly [9–11]. References [12–14] also treat the OS^2L problem as online semi-supervised clustering in that there are some *must-links* pairs (in the same cluster) and *cannot-links* pairs (cannot in the same cluster), but the effects of these methods are often influenced by “bridge points” (see a survey in [15]).

Coregularization [2, 16] is a method of improving the generalization accuracy of SVMs [17] by using unlabeled data in different views. Multiple hypotheses are trained in coregularization framework and are required to make similar predictions on any given unlabeled example. Moreover, theoretical investigations demonstrate that the coregularization approach reduces the Rademacher complexity by an amount that depends on the “distance” between the views [18, 19]. Unfortunately, basic offline coregularization algorithms are still unable to deal with long-playing large-scale OS^2L problems directly because of the constraint of time and memory.

In this paper, we introduce a novel online coregularization framework for the design and analysis of new OS^2L

algorithms. Since decreasing the primal coregularization objective function is impossible before obtaining all the training examples, we propose a Fenchel conjugate transform to increase the dual problem incrementally. The existing online coregularization algorithms in previous work can be viewed as an approximation of the dual ascending process based on gradient ascent. New online coregularization algorithms are derived based on the idea of ascending the dual function more aggressively. We also discuss the applicability of our framework to the settings where the target hypothesis is not fixed but drifts with the sequence of examples.

To the best of our knowledge, the closest prior work is proposed by de Ruijter and Tsivtsivadze [10]. Their method defines an instantaneous regularized risk function using part of examples to avoid optimizing the primal coregularization problem directly. The learning process is based on convex programming with stochastic gradient descent in kernel space. The update scheme of this work can also be derived from our online coregularization framework.

The rest of the paper will be organized as follows. In Section 2 we begin with a primal view of multiview semi-supervised learning problem based on coregularization. In Section 3 our new framework for designing and analyzing online coregularization algorithms is introduced. Next, in Section 4, we demonstrate that the existing online coregularization algorithms can be derived from our framework using gradient ascent. New online coregularization algorithms are derived based on aggressive dual ascending procedures in Section 5. Experiments and analyses are in Section 6. In Section 7, conclusions and possible extensions of our work are given.

2. Basic Problem Setting

Our notation and problem setting are formally introduced in this section. The italic lower case letters refer to scalars (e.g., α and w), and the bold letters refer to vectors (e.g., ω and λ). $(\mathbf{x}_t, y_t, \sigma_t)$ denotes the t th training example, where $\mathbf{x}_t = (\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)}, \dots, \mathbf{x}_t^{(s)})$ is seen in s views with $\mathbf{x}_t^{(i)} \in X^{(i)}$ ($i \in \{1, 2, \dots, s\}$), y_t is its label, and σ_t is a flag to determine whether the label can be seen. If $\sigma = 1$, the example is labeled; and if $\sigma = 0$, the example is unlabeled. The hinge function is denoted by $[a]_+ = \max\{a, 0\}$. $\langle \omega, \mathbf{x} \rangle$ denotes the inner product between vectors ω and $\mathbf{x}$.

In most cases, the hypotheses used for prediction come from a parameterized set of hypotheses, $\mathcal{H} = h_\omega : \omega \in V$, where V is a subset of a vector space. For simplicity and concreteness, we focus on semi-supervised binary classification problems in two views in this paper. Consider an input sequence $(\mathbf{x}_1^{(1,2)}, y_1, \sigma_1), (\mathbf{x}_2^{(1,2)}, y_2, \sigma_2), \dots, (\mathbf{x}_T^{(1,2)}, y_T, \sigma_T)$, where $\mathbf{x}_t^{(1,2)} = (\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)})$, $y_t \in \{-1, +1\}$, and $\sigma_t \in \{0, 1\}$ ($t \in \{1, 2, \dots, T\}$). The goal is to learn the function pair $f = (f^{(1)}, f^{(2)})$, where $f^{(1)} = \langle \omega^{(1)}, \mathbf{x}^{(1)} \rangle$ and $f^{(2)} = \langle \omega^{(2)}, \mathbf{x}^{(2)} \rangle$ are linear classifiers in two views, respectively. To learn max-margin decision boundaries, the

multiview S^2L problem based on coregularization [16] can be written as minimizing:

$$\begin{aligned} J(\omega^{(1)}, \omega^{(2)}) &= \frac{1}{2} \gamma_1 \|\omega^{(1)}\|^2 + \frac{1}{2} \gamma_2 \|\omega^{(2)}\|^2 \\ &+ \sum_{t=1}^T \sigma_t \left([1 - y_t \langle \omega^{(1)}, \mathbf{x}_t^{(1)} \rangle]_+ + \mu [1 - y_t \langle \omega^{(2)}, \mathbf{x}_t^{(2)} \rangle]_+ \right) \\ &+ \sum_{t=1}^T (1 - \sigma_t) \gamma_c d(\langle \omega^{(1)}, \mathbf{x}_t^{(1)} \rangle, \langle \omega^{(2)}, \mathbf{x}_t^{(2)} \rangle), \end{aligned} \quad (1)$$

where γ_1 and γ_2 are trade-off parameters for complexity functions in the two views, μ is a real-valued parameter to balance data fitting, γ_c is the coupling parameter that regularizes the pair towards compatibility using unlabeled data, and $d(\cdot, \cdot)$ is the distance function which measures the difference between the predictions for the same example and composes the multiview coregularizer.

In previous approaches based on coregularization [16, 19], the distance function $d(\cdot, \cdot)$ is often defined as a square function:

$$d(\langle \omega^{(1)}, \mathbf{x}_t^{(1)} \rangle, \langle \omega^{(2)}, \mathbf{x}_t^{(2)} \rangle) = (\langle \omega^{(1)}, \mathbf{x}_t^{(1)} \rangle - \langle \omega^{(2)}, \mathbf{x}_t^{(2)} \rangle)^2. \quad (2)$$

The distance function is defined as an absolute function (using l_1 norm) in this paper (this idea is also adopted by Szedmak and Shawe-Taylor [18] and Sun et al. [20]):

$$d(\langle \omega^{(1)}, \mathbf{x}_t^{(1)} \rangle, \langle \omega^{(2)}, \mathbf{x}_t^{(2)} \rangle) = |\langle \omega^{(1)}, \mathbf{x}_t^{(1)} \rangle - \langle \omega^{(2)}, \mathbf{x}_t^{(2)} \rangle|. \quad (3)$$

Furthermore, (3) is composed of two hinge functions (see Figure 1 for an illustration)

$$\begin{aligned} d &= [\langle \omega^{(1)}, \mathbf{x}_t^{(1)} \rangle - \langle \omega^{(2)}, \mathbf{x}_t^{(2)} \rangle]_+ \\ &+ [\langle \omega^{(2)}, \mathbf{x}_t^{(2)} \rangle - \langle \omega^{(1)}, \mathbf{x}_t^{(1)} \rangle]_+. \end{aligned} \quad (4)$$

In the next section, we will show that the online coregularization problem can be discussed in the dual form of (1) more easily and directly while using the absolute distance function.

Denote the *instantaneous loss* on round t as

$$\begin{aligned} g_t(\omega^{(1)}, \omega^{(2)}) &= \sigma_t \left([1 - y_t \langle \omega^{(1)}, \mathbf{x}_t^{(1)} \rangle]_+ \right. \\ &\quad \left. + \mu [1 - y_t \langle \omega^{(2)}, \mathbf{x}_t^{(2)} \rangle]_+ \right) \\ &+ (1 - \sigma_t) \gamma_c |\langle \omega^{(1)}, \mathbf{x}_t^{(1)} \rangle - \langle \omega^{(2)}, \mathbf{x}_t^{(2)} \rangle|, \end{aligned} \quad (5)$$

where $t \in \{1, 2, \dots, T\}$. We thus get a simple version of (1) using (5):

$$\begin{aligned} J(\omega^{(1)}, \omega^{(2)}) &= \frac{1}{2} \gamma_1 \|\omega^{(1)}\|^2 + \frac{1}{2} \gamma_2 \|\omega^{(2)}\|^2 \\ &+ \sum_{t=1}^T g_t(\omega^{(1)}, \omega^{(2)}). \end{aligned} \quad (6)$$

FIGURE 1: Two different distance functions and hinge functions. The top plane shows the two different distance function. The bottom plane shows that the absolute function $|x|$ can be decomposed into two hinge functions $[x]_+$ and $[-x]_+$.

The minimization problem of (6) in an online manner is what we consider in the rest of this paper.

3. Online Coregularization by Ascending the Dual Function

In this section, we propose a unified online coregularization framework for multi-view semi-supervised binary classification problems. Our presentation reveals how the multiview S^2L problem based on coregularization in Section 2 can be optimized in an online manner.

Before describing our framework, let us recall the definition of Fenchel conjugate that we use as a main analysis tool in this paper (see the appendix for more details). The Fenchel conjugate of a function $f: \text{dom } f \rightarrow \mathbb{R}$ is defined as

$$f^*(\lambda) = \sup \{ \langle \lambda, \omega \rangle - f(\omega) : \omega \in \text{dom } f \}. \quad (7)$$

As shown in (7), the Fenchel conjugate is defined only for single variable function in former convex analysis. We extend the definition of Fenchel conjugate to multivariable functions for solving online multiview S^2L problem in this paper.

The Fenchel conjugate of a multivariable function $f(\omega^{(1)}, \omega^{(2)}, \dots, \omega^{(s)})$ ($\omega^{(1)} \in X^{(1)}, \omega^{(2)} \in X^{(2)}, \dots, \omega^{(s)} \in X^{(s)}$) is defined as

$$\begin{aligned} f^*(\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(s)}) \\ = \sup \left\{ \sum_{i=1}^s \langle \lambda^{(i)}, \omega^{(i)} \rangle - f(\omega^{(1)}, \omega^{(2)}, \dots, \omega^{(s)}) : \right. \\ \left. \omega^{(1)} \in X^{(1)}, \omega^{(2)} \in X^{(2)}, \dots, \omega^{(s)} \in X^{(s)} \right\}. \end{aligned} \quad (8)$$

Specially, the Fenchel conjugate of hinge functions is a key to transfer coregularization from offline to online in this paper.

Proposition 1. Let $f(\omega^{(1)}, \omega^{(2)}) = [b - (\langle \omega^{(1)}, \mathbf{x}^{(1)} \rangle - \langle \omega^{(2)}, \mathbf{x}^{(2)} \rangle)]_+$, where $\omega^{(1)} \in X^{(1)}, \omega^{(2)} \in X^{(2)}$, and $b \in \mathbb{R}$. The Fenchel conjugate of $f(\omega^{(1)}, \omega^{(2)})$ is

$$f^*(\lambda^{(1)}, \lambda^{(2)}) = \begin{cases} -\alpha b & \text{if } \lambda^{(1)} = -\alpha \mathbf{x}^{(1)}, \lambda^{(2)} = \alpha \mathbf{x}^{(2)}, \\ & \text{where } \alpha \in [0, 1] \\ \infty & \text{otherwise.} \end{cases} \quad (9)$$

Proof. We first rewrite the $f(\omega^{(1)}, \omega^{(2)})$ as follows:

$$\begin{aligned} f(\omega^{(1)}, \omega^{(2)}) &= [b - (\langle \omega^{(1)}, \mathbf{x}^{(1)} \rangle - \langle \omega^{(2)}, \mathbf{x}^{(2)} \rangle)]_+ \\ &= \max_{\alpha \in [0, 1]} \alpha (b - (\langle \omega^{(1)}, \mathbf{x}^{(1)} \rangle - \langle \omega^{(2)}, \mathbf{x}^{(2)} \rangle)). \end{aligned} \quad (10)$$

Based on the definition of Fenchel conjugate for multivariable functions, we can obtain that

$$\begin{aligned} f^*(\lambda^{(1)}, \lambda^{(2)}) \\ &= \max_{(\omega^{(1)}, \omega^{(2)})} (\langle \lambda^{(1)}, \omega^{(1)} \rangle + \langle \lambda^{(2)}, \omega^{(2)} \rangle - f(\omega^{(1)}, \omega^{(2)})) \\ &= \max_{(\omega^{(1)}, \omega^{(2)})} \left(\langle \lambda^{(1)}, \omega^{(1)} \rangle + \langle \lambda^{(2)}, \omega^{(2)} \rangle \right. \\ &\quad \left. - \max_{\alpha \in [0, 1]} \alpha (b - (\langle \omega^{(1)}, \mathbf{x}^{(1)} \rangle - \langle \omega^{(2)}, \mathbf{x}^{(2)} \rangle)) \right) \end{aligned}$$

$$\begin{aligned}
&= \max_{(\boldsymbol{\omega}^{(1)}, \boldsymbol{\omega}^{(2)})} \min_{\alpha \in [0,1]} (\langle \boldsymbol{\lambda}^{(1)}, \boldsymbol{\omega}^{(1)} \rangle + \langle \boldsymbol{\lambda}^{(2)}, \boldsymbol{\omega}^{(2)} \rangle \\
&\quad - \alpha (b - (\langle \boldsymbol{\omega}^{(1)}, \mathbf{x}^{(1)} \rangle - \langle \boldsymbol{\omega}^{(2)}, \mathbf{x}^{(2)} \rangle))) \\
&= \min_{\alpha \in [0,1]} \max_{(\boldsymbol{\omega}^{(1)}, \boldsymbol{\omega}^{(2)})} (-\alpha b + \langle \boldsymbol{\lambda}^{(1)} + \alpha \mathbf{x}^{(1)}, \boldsymbol{\omega}^{(1)} \rangle \\
&\quad + \langle \boldsymbol{\lambda}^{(2)} - \alpha \mathbf{x}^{(2)}, \boldsymbol{\omega}^{(2)} \rangle) \\
&= \min_{\alpha \in [0,1]} \left(-\alpha b + \max_{\boldsymbol{\omega}^{(1)}} \langle \boldsymbol{\lambda}^{(1)} + \alpha \mathbf{x}^{(1)}, \boldsymbol{\omega}^{(1)} \rangle \right. \\
&\quad \left. + \max_{\boldsymbol{\omega}^{(2)}} \langle \boldsymbol{\lambda}^{(2)} - \alpha \mathbf{x}^{(2)}, \boldsymbol{\omega}^{(2)} \rangle \right). \tag{11}
\end{aligned}$$

Since the previous third equality follows from the strong max-min property, it can be transferred into a min-max problem. If $\boldsymbol{\lambda}^{(1)} + \alpha \mathbf{x}^{(1)} \neq 0$, $\max_{\boldsymbol{\omega}^{(1)}} \langle \boldsymbol{\lambda}^{(1)} + \alpha \mathbf{x}^{(1)}, \boldsymbol{\omega}^{(1)} \rangle$ is ∞ , and if $\boldsymbol{\lambda}^{(2)} - \alpha \mathbf{x}^{(2)} \neq 0$, $\max_{\boldsymbol{\omega}^{(2)}} \langle \boldsymbol{\lambda}^{(2)} - \alpha \mathbf{x}^{(2)}, \boldsymbol{\omega}^{(2)} \rangle$ is ∞ ; otherwise, if $\boldsymbol{\lambda}^{(1)} + \alpha \mathbf{x}^{(1)} = 0$ and $\boldsymbol{\lambda}^{(2)} - \alpha \mathbf{x}^{(2)} = 0$, we have $f^*(\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}) = -\alpha b$. $\square$

Back to the primal problem, we want to get a sequence of boundary $\boldsymbol{\omega}_0^{(1,2)}, \boldsymbol{\omega}_1^{(1,2)}, \dots, \boldsymbol{\omega}_T^{(1,2)}$ ($\boldsymbol{\omega}_t^{(1,2)} = (\boldsymbol{\omega}_t^{(1)}, \boldsymbol{\omega}_t^{(2)})$, $t \in \{0, 1, \dots, T\}$) which makes $J(\boldsymbol{\omega}_0^{(1,2)}) \geq J(\boldsymbol{\omega}_1^{(1,2)}) \geq \dots \geq J(\boldsymbol{\omega}_T^{(1,2)})$. Unfortunately, decreasing the objective function $J(\boldsymbol{\omega}^{(1)}, \boldsymbol{\omega}^{(2)})$ directly is impossible, while the training examples arrive in a stream. In order to avoid this contradiction, we propose a Fenchel conjugate transform for multi-view S^2L problems based on coregularization.

An equivalent problem of (6) is

$$\begin{aligned}
&\min_{\boldsymbol{\omega}_0^{(1,2)}, \boldsymbol{\omega}_1^{(1,2)}, \dots, \boldsymbol{\omega}_T^{(1,2)}} \frac{1}{2} \gamma_1 \|\boldsymbol{\omega}_0^{(1)}\|^2 + \frac{1}{2} \gamma_2 \|\boldsymbol{\omega}_0^{(2)}\|^2 + \sum_{t=1}^T g_t(\boldsymbol{\omega}_t^{(1,2)}), \\
&\text{s.t.} \quad \forall i \in 1, 2, \dots, T, \boldsymbol{\omega}_i^{(1)} = \boldsymbol{\omega}_0^{(1)}, \boldsymbol{\omega}_i^{(2)} = \boldsymbol{\omega}_0^{(2)}. \tag{12}
\end{aligned}$$

Using the Lagrange dual function, we can rewrite (12) by introducing a vector group $(\boldsymbol{\lambda}_1^{(1,2)}, \boldsymbol{\lambda}_2^{(1,2)}, \dots, \boldsymbol{\lambda}_T^{(1,2)})$:

$$\begin{aligned}
&\max_{\boldsymbol{\lambda}_1^{(1,2)}, \boldsymbol{\lambda}_2^{(1,2)}, \dots, \boldsymbol{\lambda}_T^{(1,2)}} \min_{\boldsymbol{\omega}_0^{(1,2)}, \boldsymbol{\omega}_1^{(1,2)}, \dots, \boldsymbol{\omega}_T^{(1,2)}} \left(\frac{1}{2} \gamma_1 \|\boldsymbol{\omega}_0^{(1)}\|^2 + \frac{1}{2} \gamma_2 \|\boldsymbol{\omega}_0^{(2)}\|^2 \right. \\
&\quad + \sum_{t=1}^T g_t(\boldsymbol{\omega}_t^{(1,2)}) \\
&\quad + \sum_{t=1}^T \langle \boldsymbol{\lambda}_t^{(1)}, \boldsymbol{\omega}_0^{(1)} - \boldsymbol{\omega}_t^{(1)} \rangle \\
&\quad \left. + \sum_{t=1}^T \langle \boldsymbol{\lambda}_t^{(2)}, \boldsymbol{\omega}_0^{(2)} - \boldsymbol{\omega}_t^{(2)} \rangle \right). \tag{13}
\end{aligned}$$

Consider the dual function

$$\begin{aligned}
&D(\boldsymbol{\lambda}_1^{(1,2)}, \boldsymbol{\lambda}_2^{(1,2)}, \dots, \boldsymbol{\lambda}_T^{(1,2)}) \\
&= \min_{\boldsymbol{\omega}_0^{(1,2)}, \boldsymbol{\omega}_1^{(1,2)}, \dots, \boldsymbol{\omega}_T^{(1,2)}} \left(\frac{1}{2} \gamma_1 \|\boldsymbol{\omega}_0^{(1)}\|^2 + \frac{1}{2} \gamma_2 \|\boldsymbol{\omega}_0^{(2)}\|^2 \right. \\
&\quad + \sum_{t=1}^T g_t(\boldsymbol{\omega}_t^{(1,2)}) \\
&\quad + \sum_{t=1}^T \langle \boldsymbol{\lambda}_t^{(1)}, \boldsymbol{\omega}_0^{(1)} - \boldsymbol{\omega}_t^{(1)} \rangle \\
&\quad \left. + \sum_{t=1}^T \langle \boldsymbol{\lambda}_t^{(2)}, \boldsymbol{\omega}_0^{(2)} - \boldsymbol{\omega}_t^{(2)} \rangle \right) \\
&= -\max_{\boldsymbol{\omega}_0^{(1)}} \left(\sum_{t=1}^T \langle -\boldsymbol{\lambda}_t^{(1)}, \boldsymbol{\omega}_0^{(1)} \rangle - \frac{1}{2} \gamma_1 \|\boldsymbol{\omega}_0^{(1)}\|^2 \right) \\
&\quad - \max_{\boldsymbol{\omega}_0^{(2)}} \left(\sum_{t=1}^T \langle -\boldsymbol{\lambda}_t^{(2)}, \boldsymbol{\omega}_0^{(2)} \rangle - \frac{1}{2} \gamma_2 \|\boldsymbol{\omega}_0^{(2)}\|^2 \right) \\
&\quad - \sum_{t=1}^T \max_{\boldsymbol{\lambda}_t^{(1,2)}} (\langle \boldsymbol{\lambda}_t^{(1)}, \boldsymbol{\omega}_t^{(1)} \rangle + \langle \boldsymbol{\lambda}_t^{(2)}, \boldsymbol{\omega}_t^{(2)} \rangle - g_t(\boldsymbol{\omega}_t^{(1,2)})) \\
&= -\frac{1}{2\gamma_1} \left(-\sum_{t=1}^T \boldsymbol{\lambda}_t^{(1)} \right)^2 \\
&\quad - \frac{1}{2\gamma_2} \left(-\sum_{t=1}^T \boldsymbol{\lambda}_t^{(2)} \right)^2 - \sum_{t=1}^T g_t^*(\boldsymbol{\lambda}_t^{(1,2)}), \tag{14}
\end{aligned}$$

where $g_t^*(\boldsymbol{\lambda}_t^{(1,2)})$ is the Fenchel conjugate of $g_t(\boldsymbol{\omega}_t^{(1,2)})$. The primal problem can be described as maximizing the dual function as in the following

$$\min_{\boldsymbol{\omega}^{(1,2)}} J(\boldsymbol{\omega}^{(1,2)}) = \max_{\boldsymbol{\lambda}_1^{(1,2)}, \boldsymbol{\lambda}_2^{(1,2)}, \dots, \boldsymbol{\lambda}_T^{(1,2)}} D(\boldsymbol{\lambda}_1^{(1,2)}, \boldsymbol{\lambda}_2^{(1,2)}, \dots, \boldsymbol{\lambda}_T^{(1,2)}). \tag{15}$$

Based on our definition of Fenchel conjugate for multi-variables functions, the Fenchel conjugate of $g_t(\boldsymbol{\omega}_t^{(1,2)})$ can be rewritten as (based on Proposition 1 and Lemma A.2 in the appendix)

$$\begin{aligned}
&g_t^*(\boldsymbol{\lambda}_t^{(1,2)}) \\
&= \begin{cases} -\sigma_t(\alpha_{t1} + \mu\alpha_{t2}), & \text{if } \boldsymbol{\lambda}_t^{(1)} = -\sigma_t\gamma_t\alpha_{t1}\mathbf{x}_t^{(1)} - (1-\sigma_t)\alpha_{t0}\gamma_c\mathbf{x}_t^{(1)}, \\ & \boldsymbol{\lambda}_t^{(2)} = -\sigma_t\mu\gamma_t\alpha_{t2}\mathbf{x}_t^{(2)} + (1-\sigma_t)\alpha_{t0}\gamma_c\mathbf{x}_t^{(2)}, \\ & \text{where } \alpha_{t1}, \alpha_{t2} \in [0, 1], \alpha_{t0} \in [-1, 1]; \\ \infty, & \text{otherwise.} \end{cases} \tag{16}
\end{aligned}$$

Since our goal is to maximize the dual function, we can restrict to the first case in (16). $g_t^*(\boldsymbol{\lambda}_t^{(1,2)})$ has 3 associated coefficient variables which are α_{t0} , α_{t1} , and α_{t2} .

INPUT: four positive scalars: $\gamma_1, \gamma_2, \gamma_c$ and μ .
 INITIALIZE: a coefficient vectors group $(\alpha_1)_0, (\alpha_2)_0, \dots, (\alpha_T)_0$ and its associated decision boundary vectors $\omega_0^{(1)}, \omega_0^{(2)}$.
 PROCESS: For $t = 1, 2, \dots, T$
     Receive an example $(\mathbf{x}_t^{(1,2)}, y_t, \sigma_t)$,
     Choose a new coefficient vectors group $(\alpha_1)_t, (\alpha_2)_t, \dots, (\alpha_T)_t$ that satisfies
          $D((\alpha_1)_t, (\alpha_2)_t, \dots, (\alpha_T)_t) \geq D((\alpha_1)_{t-1}, (\alpha_2)_{t-1}, \dots, (\alpha_T)_{t-1})$,
     Return two new associated decision boundary vectors $\omega_t^{(1)}, \omega_t^{(2)}$ using (19).

ALGORITHM 1: A template online co-regularization algorithm for multiview semi-supervised binary classification problems. This template algorithm aims for increasing the dual function on each learning round.

Based on the previous analysis, the dual function can be rewritten using a new coefficient vectors group $\alpha_1, \alpha_2, \dots, \alpha_T$, where $\alpha_t = [\alpha_{t0}, \alpha_{t1}, \alpha_{t2}]$ ($t \in \{1, 2, \dots, T\}$). Consider the following:

$$\begin{aligned}
 D(\alpha_1, \alpha_2, \dots, \alpha_T) &= -\frac{1}{2\gamma_1} \left(\sum_{t=1}^T (\sigma_t y_t \alpha_{t1} \mathbf{x}_t^{(1)} + (1 - \sigma_t) \alpha_{t0} \gamma_c \mathbf{x}_t^{(1)}) \right)^2 \\
 &\quad - \frac{1}{2\gamma_2} \left(\sum_{t=1}^T (\sigma_t \mu y_t \alpha_{t2} \mathbf{x}_t^{(2)} - (1 - \sigma_t) \alpha_{t0} \gamma_c \mathbf{x}_t^{(2)}) \right)^2 \quad (17) \\
 &\quad + \sum_{t=1}^T \sigma_t (\alpha_{t1} + \mu \alpha_{t2}).
 \end{aligned}$$

As shown in (17), our task has been transferred to a constrained quadratic programming (QP) optimization problem. Every input training example brings a vector α into the dual function. $\alpha_1, \alpha_2, \dots, \alpha_T$ are independent, so we can update the vectors group $(\alpha_1, \alpha_2, \dots, \alpha_T)$ on each learning round to ascend the dual problem incrementally. Obviously, unobserved examples would make no influence on the value of dual function in (17) by setting their associated coefficient variables to zero.

Denote $(\alpha_i)_t$ the coefficient vector α_i on round t ($i \in \{1, 2, \dots, T\}$). The update process of coefficient vectors group $(\alpha_1, \alpha_2, \dots, \alpha_T)$ on round t should satisfy the following conditions:

- (i) if $t + 1 \leq i \leq T$, $(\alpha_{i0})_t, (\alpha_{i1})_t, (\alpha_{i2})_t = 0$;
- (ii) $D((\alpha_1)_t, (\alpha_2)_t, \dots, (\alpha_T)_t) \geq D((\alpha_1)_{t-1}, (\alpha_2)_{t-1}, \dots, (\alpha_T)_{t-1})$.

The first one means that the unobserved examples do not make influence on the value of dual function, and the second means that the value of dual function never decreases during the online coregularization process. Therefore, the dual function on round t can also be written as

$$\begin{aligned}
 D((\alpha_1)_t, (\alpha_2)_t, \dots, (\alpha_T)_t) &= -\frac{1}{2\gamma_1} \left(\sum_{i=1}^t (\sigma_i y_i (\alpha_{i1})_t \mathbf{x}_i^{(1)} + (1 - \sigma_i) (\alpha_{i0})_t \gamma_c \mathbf{x}_i^{(1)}) \right)^2
 \end{aligned}$$

$$\begin{aligned}
 &- \frac{1}{2\gamma_2} \left(\sum_{i=1}^t (\sigma_i y_i \mu (\alpha_{i2})_t \mathbf{x}_i^{(2)} - (1 - \sigma_i) (\alpha_{i0})_t \gamma_c \mathbf{x}_i^{(2)}) \right)^2 \\
 &+ \sum_{i=1}^t \sigma_i ((\alpha_{i1})_t + \mu (\alpha_{i2})_t). \quad (18)
 \end{aligned}$$

Based on Lemmas A.1 and A.3 in the appendix, we can obtain that each coefficient vectors group $(\alpha_1, \alpha_2, \dots, \alpha_T)$ has an associated boundary vector group $(\omega^{(1)}, \omega^{(2)})$. On round t , the associated boundaries of $((\alpha_1)_t, (\alpha_2)_t, \dots, (\alpha_T)_t)$ are

$$\begin{aligned}
 \omega_t^{(1)} &= \frac{1}{\gamma_1} \sum_{i=1}^t (\sigma_i y_i (\alpha_{i1})_t \mathbf{x}_i^{(1)} + (1 - \sigma_i) (\alpha_{i0})_t \gamma_c \mathbf{x}_i^{(1)}), \\
 \omega_t^{(2)} &= \frac{1}{\gamma_2} \sum_{i=1}^t (\sigma_i \mu y_i (\alpha_{i2})_t \mathbf{x}_i^{(2)} - (1 - \sigma_i) (\alpha_{i0})_t \gamma_c \mathbf{x}_i^{(2)}). \quad (19)
 \end{aligned}$$

To make a summary, we propose a template online coregularization algorithm by dual ascending procedure in Algorithm 1.

Essentially, our online coregularization framework aims to break the large QP in the primal objective function into a series of dual ascending procedures on each learning round. Therefore, we can ascend the dual function in an online manner.

4. Analysis of Previous Work Based on Gradient Ascent in the Dual

In the previous section, a template algorithm framework for online coregularization is proposed based on the idea of ascending the dual function. In Algorithm 1, we can obtain that algorithms that derive from our framework may vary in one of two ways. First, different algorithms may update different dual variables on each learning round. The second way in which different algorithms may vary is how to update the chosen variables to ascend the dual function.

Some online coregularization algorithms [9, 10] have been suggested in recent years. These approaches have a similar idea of “defining an instantaneous coregularized risk to

avoid optimizing the primal coregularization problem directly.” In these works, there are two popular instantaneous coregularized risk functions $J_t(\mathbf{w}^{(1)}, \mathbf{w}^{(2)})$ which are defined as

$$\begin{aligned} J_t(\mathbf{w}^{(1)}, \mathbf{w}^{(2)}) &= \sigma_t \left([1 - \gamma_t \langle \mathbf{w}^{(1)}, \mathbf{x}_t^{(1)} \rangle]_+ + \mu [1 - \gamma_t \langle \mathbf{w}^{(2)}, \mathbf{x}_t^{(2)} \rangle]_+ \right) \\ &\quad + (1 - \sigma_t) \gamma_c d(\langle \mathbf{w}^{(1)}, \mathbf{x}_t^{(1)} \rangle, \langle \mathbf{w}^{(2)}, \mathbf{x}_t^{(2)} \rangle), \\ J_t(\mathbf{w}^{(1)}, \mathbf{w}^{(2)}) &= \frac{1}{2} \gamma_1 \|\mathbf{w}^{(1)}\|^2 + \frac{1}{2} \gamma_2 \|\mathbf{w}^{(2)}\|^2 \\ &\quad + \sigma_t \left([1 - \gamma_t \langle \mathbf{w}^{(1)}, \mathbf{x}_t^{(1)} \rangle]_+ + \mu [1 - \gamma_t \langle \mathbf{w}^{(2)}, \mathbf{x}_t^{(2)} \rangle]_+ \right) \\ &\quad + (1 - \sigma_t) \gamma_c d(\langle \mathbf{w}^{(1)}, \mathbf{x}_t^{(1)} \rangle, \langle \mathbf{w}^{(2)}, \mathbf{x}_t^{(2)} \rangle). \end{aligned} \quad (20)$$

The online coregularization process in these works is based on convex programming with gradient descent on instantaneous coregularized risk function in kernel space. The step size ρ_t is often defined to decay at a certain rate [11], for example, $\rho_t = 1/\sqrt{t}$. The update process in these approaches can be summarized as

$$\mathbf{w}_t^{(1,2)} = \mathbf{w}_{t-1}^{(1,2)} - \rho_t \frac{\partial J_t(\mathbf{w}^{(1,2)})}{\partial \mathbf{w}^{(1,2)}} \bigg|_{\mathbf{w}_{t-1}^{(1,2)}}. \quad (21)$$

In the following, we demonstrate that these algorithms can be derived from our online coregularization framework. Since the dual coefficient vectors $\alpha_1, \alpha_2, \dots, \alpha_T$ are independent, the dual function can be ascended by updating only the associated coefficient vector α_t of the new arrived training example $(\mathbf{x}_t^{(1,2)}, y_t, \sigma_t)$ on round t that means

$$(\alpha_i)_t = (\alpha_i)_{t-1}, \quad \text{if } i \neq t. \quad (22)$$

And the task on round t can be rewritten as ascending

$$\begin{aligned} D_t(\alpha_t) &= -\frac{1}{2\gamma_1} (\gamma_1 \mathbf{w}_{t-1}^{(1)} + \sigma_t y_t \alpha_{t1} \mathbf{x}_t^{(1)} + (1 - \sigma_t) \alpha_{t0} \gamma_c \mathbf{x}_t^{(1)})^2 \\ &\quad - \frac{1}{2\gamma_2} (\gamma_2 \mathbf{w}_{t-1}^{(2)} + \sigma_t \mu y_t \alpha_{t2} \mathbf{x}_t^{(2)} - (1 - \sigma_t) \alpha_{t0} \gamma_c \mathbf{x}_t^{(2)})^2 \\ &\quad + \sum_{i=1}^t \sigma_i ((\alpha_{i1})_t + \mu (\alpha_{i2})_t). \end{aligned} \quad (23)$$

Using a gradient ascent (GA) step on α_t , the update process on round t can be written as

$$\begin{aligned} (\alpha_{t0})_t &= \rho_t \frac{\partial D_t(\alpha_t)}{\partial \alpha_{t0}} \bigg|_{\alpha_t=0} \\ &= -\rho_t (1 - \sigma_t) \gamma_c (\langle \mathbf{w}_{t-1}^{(1)}, \mathbf{x}_t^{(1)} \rangle - \langle \mathbf{w}_{t-1}^{(2)}, \mathbf{x}_t^{(2)} \rangle), \\ (\alpha_{t1})_t &= \rho_t \frac{\partial D_t(\alpha_t)}{\partial \alpha_{t1}} \bigg|_{\alpha_t=0} = \rho_t \sigma_t [1 - \gamma_t \langle \mathbf{w}_{t-1}^{(1)}, \mathbf{x}_t^{(1)} \rangle]_+, \\ (\alpha_{t2})_t &= \rho_t \frac{\partial D_t(\alpha_t)}{\partial \alpha_{t2}} \bigg|_{\alpha_t=0} = \rho_t \sigma_t \mu [1 - \gamma_t \langle \mathbf{w}_{t-1}^{(2)}, \mathbf{x}_t^{(2)} \rangle]_+, \end{aligned} \quad (24)$$

where $\rho_t \geq 0$ is a step size.

In fact, the dual coefficient vectors in $\mathbf{w}_{t-1}^{(1,2)}$ can also be updated in (23). Since $\mathbf{w}_{t-1}^{(1,2)}$ has $t-1$ dual coefficient vectors, it is impossible to update them, respectively. We introduce a new variable η_t into (23),

$$\begin{aligned} D_t(\alpha_t, \eta_t) &= -\frac{1}{2\gamma_1} (\gamma_1 (1 - \eta_t) \mathbf{w}_{t-1}^{(1)} + \sigma_t y_t \alpha_{t1} \mathbf{x}_t^{(1)} \\ &\quad + (1 - \sigma_t) \alpha_{t0} \gamma_c \mathbf{x}_t^{(1)})^2 \\ &\quad - \frac{1}{2\gamma_2} (\gamma_2 (1 - \eta_t) \mathbf{w}_{t-1}^{(2)} + \sigma_t \mu y_t \alpha_{t2} \mathbf{x}_t^{(2)} \\ &\quad - (1 - \sigma_t) \alpha_{t0} \gamma_c \mathbf{x}_t^{(2)})^2 \\ &\quad + \sum_{i=1}^{t-1} \sigma_i ((1 - \eta_t) (\alpha_{i1})_t + \mu (1 - \eta_t) (\alpha_{i2})_t) \\ &\quad + \sigma_t ((\alpha_{t1})_t + \mu (\alpha_{t2})_t). \end{aligned} \quad (25)$$

From (25), we can get that a gradient ascent update on η_t actually means to multiply all the dual coefficient vectors in $\mathbf{w}_{t-1}^{(1,2)}$ by $1 - \eta_t$. Since every dual coefficient variable in $\mathbf{w}_{t-1}^{(1,2)}$ is constrained, we also constrain $\eta_t \in [0, 1]$. The initial value of η_t is zero. While using a gradient ascent on η_t , we obtain that

$$\begin{aligned} \eta_t &= \rho_t \left[\gamma_1 \langle \mathbf{w}_{t-1}^{(1)}, \mathbf{w}_{t-1}^{(1)} \rangle + \gamma_2 \langle \mathbf{w}_{t-1}^{(2)}, \mathbf{w}_{t-1}^{(2)} \rangle \right. \\ &\quad \left. - \sum_{i=1}^{t-1} \sigma_i (\alpha_{i1})_t - \sum_{i=1}^{t-1} \sigma_i \mu (\alpha_{i2})_t \right]_+. \end{aligned} \quad (26)$$

Based on the previous analysis, the gradient ascent process of boundary vector group $(\omega^{(1)}, \omega^{(2)})$ can be written as

$$\begin{aligned}\omega_t^{(1)} &= (1 - \eta_t) \omega_{t-1}^{(1)} \\ &\quad + \frac{1}{\gamma_1} (\sigma_t y_t (\alpha_{t1})_t \mathbf{x}_t^{(1)} + (1 - \sigma_t) (\alpha_{t0})_t \gamma_c \mathbf{x}_t^{(1)}), \\ \omega_t^{(2)} &= (1 - \eta_t) \omega_{t-1}^{(2)} \\ &\quad + \frac{1}{\gamma_2} (\sigma_t \mu y_t (\alpha_{t2})_t \mathbf{x}_t^{(2)} - (1 - \sigma_t) (\alpha_{t0})_t \gamma_c \mathbf{x}_t^{(2)}).\end{aligned}\quad (27)$$

As far as we know, all the existing online coregularization algorithms in previous work can be viewed as an approximation of our dual ascending process using gradient ascent in (27).

5. Deriving New Algorithms Based on Aggressive Dual Ascending (ADA) Procedures

In the previous section, we show that the online coregularization algorithms in previous work can be derived from our framework. These algorithms lead to a conservative increase of the value of the dual function since they only modify a single dual vector using gradient ascent on each learning round. In fact, more aggressive online coregularization algorithms can also be derived from our framework.

In this section we describe broader and, in practice, more powerful online coregularization algorithms which increase the dual function more aggressively on each learning round. The motivation for the new algorithms is as follows. Intuitively, update schemes that yield larger increases of the dual function are likely to reach the minimal value of primal objective function faster. Thus, they are in practice likely to suffer a smaller number of mistakes.

5.1. Updating Single Dual Coefficient Vector. The update scheme we described in Section 4 for increasing the dual function modifies the associated coefficient vector of the new arrived training example which is based on gradient ascent, and all the variables in the vector share a same step size. This simple algorithm can be enhanced by solving the following optimization problem on each learning round

$$\begin{aligned}\max_{\alpha_{t1}, \alpha_{t2} \in [0,1], \alpha_{t0} \in [-1,1]} & D(\alpha_1, \alpha_2, \dots, \alpha_T) \\ \text{s.t.} & \quad \forall i \neq t, \alpha_i = (\alpha_i)_{t-1}.\end{aligned}\quad (28)$$

According to the type of the new arrived example, (28) can be solved in different ways. If the new arrived example

is labeled, we have $\sigma_t = 1$, and the task on round t can be rewritten as

$$\begin{aligned}\max_{\alpha_{t1}, \alpha_{t2} \in [0,1]} & -\frac{1}{2\gamma_1} (\gamma_1 \omega_{t-1}^{(1)} + y_t \alpha_{t1} \mathbf{x}_t^{(1)})^2 \\ & -\frac{1}{2\gamma_2} (\gamma_2 \omega_{t-1}^{(2)} + \mu y_t \alpha_{t2} \mathbf{x}_t^{(2)})^2 + (\alpha_{t1} + \mu \alpha_{t2}) \\ = & \max_{\alpha_{t1} \in [0,1]} \left(-\frac{1}{2\gamma_1} (\gamma_1 \omega_{t-1}^{(1)} + y_t \alpha_{t1} \mathbf{x}_t^{(1)})^2 + \alpha_{t1} \right) \\ & + \max_{\alpha_{t2} \in [0,1]} \left(-\frac{1}{2\gamma_2} (\gamma_2 \omega_{t-1}^{(2)} + \mu y_t \alpha_{t2} \mathbf{x}_t^{(2)})^2 + \mu \alpha_{t2} \right).\end{aligned}\quad (29)$$

Since $\alpha_{t1}, \alpha_{t2} \in [0, 1]$, we can obtain that

$$\begin{aligned}(\alpha_{t1})_t &= \min \left\{ \frac{\gamma_1 [1 - y_t \langle \omega_{t-1}^{(1)}, \mathbf{x}_t^{(1)} \rangle]_+}{\langle \mathbf{x}_t^{(1)}, \mathbf{x}_t^{(1)} \rangle}, 1 \right\}, \\ (\alpha_{t2})_t &= \min \left\{ \frac{\gamma_2 [1 - y_t \langle \omega_{t-1}^{(2)}, \mathbf{x}_t^{(2)} \rangle]_+}{\mu \langle \mathbf{x}_t^{(2)}, \mathbf{x}_t^{(2)} \rangle}, 1 \right\}.\end{aligned}\quad (30)$$

Otherwise, if the new arrived example is unlabeled, we have $\sigma_t = 0$, and (28) can be rewritten as

$$\begin{aligned}\max_{\alpha_{t0} \in [-1,1]} & -\frac{1}{2\gamma_1} (\gamma_1 \omega_{t-1}^{(1)} + \alpha_{t0} \gamma_c \mathbf{x}_t^{(1)})^2 \\ & -\frac{1}{2\gamma_2} (\gamma_2 \omega_{t-1}^{(2)} - \alpha_{t0} \gamma_c \mathbf{x}_t^{(2)})^2.\end{aligned}\quad (31)$$

Since $\alpha_{t0} \in [-1, 1]$, we can obtain that

$$\begin{aligned}(\alpha_{t0})_t &= \begin{cases} \min \left\{ -\frac{(1/\gamma_c) (\langle \omega_{t-1}^{(1)}, \mathbf{x}_t^{(1)} \rangle - \langle \omega_{t-1}^{(2)}, \mathbf{x}_t^{(2)} \rangle)}{(1/\gamma_2) \langle \mathbf{x}_t^{(2)}, \mathbf{x}_t^{(2)} \rangle + (1/\gamma_1) \langle \mathbf{x}_t^{(1)}, \mathbf{x}_t^{(1)} \rangle}, 1 \right\}, \\ \quad \text{if } (\alpha_{t0})_t \geq 0 \\ \max \left\{ -\frac{(1/\gamma_c) (\langle \omega_{t-1}^{(1)}, \mathbf{x}_t^{(1)} \rangle - \langle \omega_{t-1}^{(2)}, \mathbf{x}_t^{(2)} \rangle)}{(1/\gamma_2) \langle \mathbf{x}_t^{(2)}, \mathbf{x}_t^{(2)} \rangle + (1/\gamma_1) \langle \mathbf{x}_t^{(1)}, \mathbf{x}_t^{(1)} \rangle}, -1 \right\} \\ \quad \text{if } (\alpha_{t0})_t < 0. \end{cases}\end{aligned}\quad (32)$$

Based on the previous analysis, the update process of boundary vector group $(\omega^{(1)}, \omega^{(2)})$ can be written as

$$\begin{aligned}\omega_t^{(1)} &= \omega_{t-1}^{(1)} + \frac{1}{\gamma_1} \\ &\quad \times (\sigma_t y_t (\alpha_{t1})_t \mathbf{x}_t^{(1)} + (1 - \sigma_t) (\alpha_{t0})_t \gamma_c \mathbf{x}_t^{(1)}), \\ \omega_t^{(2)} &= \omega_{t-1}^{(2)} + \frac{1}{\gamma_2} \\ &\quad \times (\sigma_t \mu y_t (\alpha_{t2})_t \mathbf{x}_t^{(2)} - (1 - \sigma_t) (\alpha_{t0})_t \gamma_c \mathbf{x}_t^{(2)}).\end{aligned}\quad (33)$$

In contrast to the gradient approaches in Section 4, this approach ascends the dual function more aggressively. So far,

FIGURE 2: Four choices of S_t to update multiple dual coefficient vectors. The horizontal thin line on each learning round represents the whole training data sequence, while the thick boxes represent the set of examples whose dual coefficient vectors are used to ascend the dual function on that round. Essentially, different choices of S_t construct different QP problems on each learning round. Obviously, if $S_t = [t]$, the update scheme chooses to solve the largest possible QP on each learning round; and if $S_t = \{t\}$, the update scheme chooses to solve the smallest possible QP on each learning round.

our focus was on an update which modifies a single dual coefficient vector. In fact, all the associated coefficient vectors of the arrived examples can be updated during the online regularization process. We now examine another update scheme based on our online coregularization framework that suggests the modification of multiple dual coefficient vectors on each learning round.

5.2. Updating Multiple Dual Coefficient Vectors. We now describe a new update scheme which modifies multiple dual coefficient vectors on each learning round. In practice, we can get the example set $\{(\mathbf{x}_1^{(1,2)}, y_1, \sigma_1), (\mathbf{x}_2^{(1,2)}, y_2, \sigma_2), \dots, (\mathbf{x}_t^{(1,2)}, y_t, \sigma_t)\}$ on round t , and all the associated coefficient vectors of these examples can be updated to ascend the dual function.

Denote $S_t \subseteq [t]$ the buffer on round t , where $t \in S_t$ and $[t] = \{1, 2, \dots, t\}$. If $i \in S_t$, the associated coefficient vector of the i th example can be updated on round t . We optimize the dual function over the dual coefficient vectors whose indices belong to S_t . Consider the following:

$$\begin{aligned} \max_{\alpha_i (i \in S_t)} \quad & D(\alpha_1, \alpha_2, \dots, \alpha_T) \\ \text{s.t.} \quad & \forall i \notin S_t, \alpha_i = (\alpha_i)_{t-1}. \end{aligned} \quad (34)$$

Obviously, this more general update scheme could make a larger increase on each learning round. If setting $S_t = [t]$, (34) makes use of all the examples that have been observed and thus is likely to make the largest increase of the dual function. And if setting $S_t = \{t\}$, (34) degenerates in the update scheme we have described in Section 5.1. The time complexity for solving (34) is $O((2 \times \text{size of } (S_t))^3)$ (worst case).

Similar to the idea in Section 4, we can also introduce a new variable η_t into (34),

$$\begin{aligned} \max_{\alpha_i (i \in S_t), \eta_t \in [0, 1]} \quad & D(\alpha_1, \alpha_2, \dots, \alpha_T) \\ \text{s.t.} \quad & \forall i \notin S_t, \alpha_i = (1 - \eta_t) \times (\alpha_i)_{t-1}. \end{aligned} \quad (35)$$

η_t is more like a *forgetting factor* [21] to downweight the contribution of observations whose indices do not belong to S_t . When tracking the changes in the data stream, it is likely that recent observations will be more indicative of its appearance than more distant ones. Incorporating a *forgetting factor* in the online learning algorithms is a good way to

moderate the balance between old and new observations. We can also obtain that $\eta_t = 0$ ($t \in \{1, 2, \dots, T\}$) indicates that no forgetting is to occur.

For practical purpose, we test two choices of S_t to update multiple dual coefficient vectors in this paper.

- (i) Buffer-N. Let the buffer size be τ . Buffer-N replaces the oldest example $(\mathbf{x}_{t-\tau}^{(1,2)}, y_{t-\tau}, \sigma_{t-\tau})$ in the buffer with the new arrived example $(\mathbf{x}_t^{(1,2)}, y_t, \sigma_t)$ on each learning round, which means $S_t = \{t - \tau + 1, t - \tau + 2, \dots, t\}$.
- (ii) Buffer-L. This buffering strategy replaces the oldest unlabeled point in the buffer with the incoming point while keeping labeled points. The oldest labeled point is evicted from the buffer only when it is filled with labeled points.

Figure 2 shows the set of the associated coefficient vectors which are used to ascend the dual function on each learning round for different choices of S_t . Essentially, different choices of S_t construct different QP problems on each learning round.

5.3. Sparse Approximations for Kernel Representation. In practice, kernel functions are always used to find a linear classifier, like SVM. Our online coregularization framework only contains the product of two points, so we can easily introduce the kernel function into our framework. If we note K the kernel matrix such that

$$K_{ij} = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j), \quad (36)$$

$\mathbf{x}_i$ can be replaced by $\Phi(\mathbf{x}_i)$ in our framework. Therefore, we can rewrite (19) as

$$\begin{aligned} \omega_t^{(1)} &= \frac{1}{\gamma_1} \sum_{i=1}^t (\sigma_i y_i (\alpha_{i1})_t + (1 - \sigma_i) (\alpha_{i0})_t \gamma_c) \Phi^{(1)}(\mathbf{x}_i^{(1)}) \\ &= \sum_{i=1}^t (\beta_i^{(1)})_t \Phi^{(1)}(\mathbf{x}_i^{(1)}), \end{aligned} \quad (37)$$

$$\begin{aligned} \omega_t^{(2)} &= \frac{1}{\gamma_2} \sum_{i=1}^t (\sigma_i \mu y_i (\alpha_{i2})_t - (1 - \sigma_i) (\alpha_{i0})_t \gamma_c) \Phi^{(2)}(\mathbf{x}_i^{(2)}) \\ &= \sum_{i=1}^t (\beta_i^{(2)})_t \Phi^{(2)}(\mathbf{x}_i^{(2)}). \end{aligned}$$

Unfortunately, our previous derived online coregularization algorithms with kernel functions have to store the example sequence up to the current round, and the stored matrix size is $t \times t$ (worst case). For practical purpose, we present two approaches to sparsify the kernel representation of boundaries on each learning round.

Absolute Threshold. To construct a sparse representation for the boundaries, absolute threshold discards the examples whose associated coefficients $\beta^{(1,2)}$ are close to zero. Let $\varepsilon > 0$ denote the absolute threshold. When an arrived example $\mathbf{x}_i^{(1,2)}$ would not be used to update the boundaries in further learning process, $\mathbf{x}_i^{(1,2)}$ is discarded if $|\beta_i^{(1,2)}| < \varepsilon$. The examples whose indices are in S_t cannot be discarded on round t since they would be used to ascend the dual function.

k Maximal Coefficients (k -MC). Another way to sparsify the kernel representation is to keep the examples of which the absolute value of $\beta^{(1,2)}$ is the first k maximum. Similar as the absolute threshold, k -MC does not discard the examples in S_t which would be used to ascend the dual function on round t . Based on this sparse approximation, the stored matrix size on round t reduces to $(k + \text{size of } (S_t)) \times (k + \text{size of } (S_t))$.

The previous two sparse approximations are both motivated by the fact that the examples which have larger coefficients tend to exert more influence on our learned boundaries.

6. Experiments

This section presents a series of experimental results to report the effectiveness of our derived online coregularization algorithms. It is known that the performance of semi-supervised learning depends on the correctness of model assumptions. Thus, our focus is on comparing different online coregularization algorithms with multiple views, rather than different semi-supervised regularization methods.

We report experimental results on two synthetic and a real word binary classification problems. The prediction function in online coregularization algorithms are adopted as the average of the prediction functions from two views

$$f_t(\mathbf{x}^{(1,2)}) = \text{sign} \left(\frac{1}{2} (\langle \mathbf{w}_t^{(1)}, \mathbf{x}^{(1)} \rangle + \langle \mathbf{w}_t^{(2)}, \mathbf{x}^{(2)} \rangle) \right). \quad (38)$$

Based on the idea of “interested in the best performance and simply select the parameter values minimizing the error” [3], we select combinations of the parameter values on a finite grid in Table 1, and it is sufficient to perform algorithm comparisons.

The training sequences are generated randomly from each data sets. To avoid the influence of different training sequences, all results on each dataset are the average of five such trials (this idea is inspired by [11]). The error rates have ± 1 standard deviation. While using buffering strategies to update multiple dual coefficient vectors, the buffer size is fixed at 100 to avoid high computational complexity on each learning round. We implemented all the experiments using MATLAB.

TABLE 1: A finite grid of parameter values. We find the best performance of each online co-regularization algorithm on this finite grid.

Parameter	Values
γ_1	$10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3$
γ_2	$10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3$
γ_c	$10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3$
μ	$10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3$

6.1. Two-Moons-Two-Lines Synthetic Data Set. This synthetic data set is generated similarly to the toy example used in [16, 19] in which examples in two classes appear as two moons in one view and two oriented lines in another (see Figure 3 for an illustration). This data set contains 2000 examples, and only 5 examples for each class are labeled. A Gaussian and linear kernel are chosen for the two-moons and two-lines views respectively. In this data set, the offline coregularization algorithms (CoLapSVM) [16] achieve an error rate of 0.

The best performance of all the online coregularization algorithms in Section 5 is presented in Table 2. We also provide some additional details during the online coregularization process.

We compare cumulative runtime curves of online coregularization algorithms with different sparse approximation approaches in Figure 4. Online coregularization algorithms with sparse representation perform better than the basic online coregularization algorithms on the growth rate. The cumulative runtime growth curves of online coregularization algorithms with sparse approximation approaches scale only linearly, while the others scale quadratically.

We also compare the number of examples in the kernel representation of boundary vectors in two views on each learning round for different sparse approximation approaches. Figure 5 shows that only part of examples have to be stored (and computed) while using sparse approximation approaches. Online coregularization algorithms without sparse approximation approaches are time consuming and memory-consuming, and it is intractable to apply them to real-world long time tasks.

In Section 3, we have demonstrated the relationship between the primal objective function $J(\mathbf{w}_t^{(1,2)})$ and the dual function $D((\boldsymbol{\alpha}_1)_t, (\boldsymbol{\alpha}_2)_t, \dots, (\boldsymbol{\alpha}_T)_t)$. We compare the primal objective function versus the dual function on the training sequence of two-moons-two-lines data set as t increases in Figure 6. The result shows that the two curves approach each other along the online coregularization algorithms run. The value of dual function never decreases as t increases; correspondingly, the curve of primal function has a downward trend and some little fluctuations. We also observe that the curve of the primal objective function which updates multiple dual coefficient vectors on each learning round shows a more smooth downward trend and has less rapid fluctuations. This experiment supports the theory that increasing the dual problem can achieve comparable risks of the primal objective function.

We report the performance of $\mathbf{w}_t^{(1,2)}$ on the whole two-moons-two-lines data set in Figure 7. This result shows that

FIGURE 3: Distribution of two-moons-two-lines data set.

FIGURE 4: Cumulative runtime growth curves. From left to right (a) ADA ($S_t = \{t\}$); (b) ADA (Buffer- N). The curves have the similar trends for different online coregularization algorithms. Online coregularization algorithms with sparse representation perform better than the basic online coregularization algorithms on the growth rate.

the boundary vector is adjusted to be a better one along the online coregularization algorithms run. Since our algorithms adjust the decision boundary vector according to the local agreement of S_t in the two views on each learning round, the curve of the error rate is not always decreasing along the online coregularization process. It is also the reason why our

online coregularization algorithms can track the changes in the data sequence (more detail in Section 6.3). Similar as the experiments in Figure 6, we can observe that the error rate of $\omega_t^{(1,2)}$ which updates multiple dual coefficient vectors on each learning round shows a more smooth downward trend and has less rapid fluctuations.

TABLE 2: Mean test error rates on the two-moons-two-lines synthetic data set. The error rates are reported for three different sparse approximations. For gradient ascent, we choose a decaying step size $\rho_t = 0.1/\sqrt{t}$. The result shows that our derived online co-regularization algorithms achieve test accuracy comparable to offline co-regularization (CoLapSVM). The online co-regularization algorithms based on aggressive dual ascending procedures perform better than those based on gradient ascent.

Online co-regularization algorithms	Gradient ascent	ADA $S_t = \{t\}$	ADA $S_t = [t]$	ADA Buffer- N	ADA Buffer- L
Error rates (%)					
No sparse approximation	12.15	8.95	3.95	4.25	4.05
Absolute threshold ($\varepsilon = 0.05$)	10.80	9.65	3.95	4.05	4.05
k -MC ($k = 400$)	13.30	9.85	3.95	7.00	5.35

(a) Two-moons view

(b) Two-lines view

FIGURE 5: The number of examples in the kernel representation of boundary vectors in two views for different sparse approximation approaches. The online coregularization algorithm used in this experiment is ADA (Buffer- N). If no sparse representation approaches are used, the kernel representation contains all the arrived examples on each learning round. The number of examples in the kernel representation increases slowly while using an absolute threshold, and the number is at most k while using k -MC for online coregularization algorithms.

(a)

(b)

FIGURE 6: The value curves of the primal objective function and the dual function during the incremental learning process. From left to right (a) ADA ($S_t = \{t\}$) and (b) ADA (Buffer- N). The two curves approach each other as t increases.

FIGURE 7: Error rates of $\omega_t^{(1,2)}$ on the whole two-moons-two-lines data set. From left to right (a) ADA ($S_t = \{t\}$) and (b) ADA (Buffer- N). The error rate of $\omega_t^{(1,2)}$ has a downward trend, but it is not always decreasing during the online coregularization process.

TABLE 3: Mean test error rates on the web page data set. The error rates are reported for three different sparse approximations. For gradient ascent, we choose a decaying step size $\rho_t = 0.1/\sqrt{t}$. The result shows that our derived online co-regularization algorithms achieve test accuracy comparable to offline co-regularization (CoLapSVM).

Online co-regularization algorithms	Gradient ascent	ADA $S_t = \{t\}$	ADA $S_t = [t]$	ADA Buffer- N	ADA Buffer- L
Error rates (%)					
No sparse approximation	10.94	11.89	7.52	8.47	7.80
Absolute threshold ($\varepsilon = 0.05$)	11.23	11.70	7.52	8.66	7.80
k -MC ($k = 400$)	11.32	11.41	7.80	8.75	7.99

6.2. Web Page Categorization. We applied our derived online coregularization algorithms to the WebKB text classification task studied in Blum and Mitchell [2]; Sindhwani et al. [16]; Sun and Shawe-Taylor [19]. The task is to predict whether a web page is a course home page or not. The data set consists of 1051 web pages in two views (page and link) collected from the computer science department web sites at four U.S. universities: Cornell, University of Washington, University of Wisconsin, and University of Texas. The first view of the data is the textual content of a webpage itself, and the second view is all links pointing to the web page from other web pages. We preprocessed each view by removing stop words, punctuation, and numbers and then applied Porter's stemming to the text [22]. This problem has an unbalanced class distribution since there are a total of 230 course home pages and 821 noncourse. In addition, words that occur in five or fewer documents were ignored. This resulted in 2332 and 87 dimensional vectors for two views, respectively. Finally, document vectors were normalized to TDIDF features (the product of term frequency and inverse document frequency) [23]. As in [19], we randomly label 3 course and 9 noncourse examples for each class. In this experiment, the linear kernel is used for both views.

In this data set, the offline coregularization algorithms (CoLapSVM) [16] achieve an error rate of 6.32%. In Table 3, we report the best performance of all the online coregularization algorithms on the web page data set.

6.3. Rotating Two-Moons-Two-Lines Synthetic Data Set. When the underlying distributions, both $P(\mathbf{x}^{(1,2)})$ and $P(y | \mathbf{x}^{(1,2)})$, change during the course of learning, the algorithms are expected to track the changes in the data sequence. In

this subsection, we test the applicability of our framework to settings where the target hypotheses in different views are not fixed but rather drift with the sequence of examples. To demonstrate that our online coregularization algorithms can handle concept drift, we perform our experiments on rotating two-moons-two-lines data sequence. This data set contains 8000 examples, and only 1% examples for each class are labeled. Figure 8 shows that the two-moons-two-lines data set smoothly rotate 360° during the sequence, and the target boundaries in two views drift with the sequence of examples. In the rotating two-moons-two-lines data set, the points will change their true labels during the sequence, and every stationary decision boundary will have an error rate of 50% approximately. A Gaussian and linear kernel are chosen for the two rotating moons and two rotating lines view, respectively.

In Table 4, we report the best performance of all the online coregularization algorithms on the rotating two-moons-two-lines data sequence. In this experiment, we also discuss the effect of the buffer size to track the changes in the data sequence.

Obviously, when tracking the changes in the rotating two-moons-two-lines data sequence, it is likely that recent examples will be more indicative of the boundaries than more distant ones. Buffer- N updates the boundaries using the recent examples, it is also the reason why ADA (Buffer- N) performs better than the other online coregularization algorithms. We report the error rates of ADA (Buffer- N) with different buffer sizes on rotating two-moons-two-lines synthetic data sequence in Figure 9. This experiment illustrates that a suitable size of Buffer- N is able to adapt to the changing sequence and maintain a small error rate.

TABLE 4: Mean test error rates on the rotating two-moons-two-lines synthetic data sequence. The error rates are reported for three different sparse approximations. For gradient ascent, we choose a stationary step size $\rho_t = 0.1$. The result shows that our derived online co-regularization algorithms are able to track the changes in the sequence and maintain a smaller error rate compared with batch learning algorithms. Specially, ADA (Buffer- N) performs better than the other online co-regularization algorithms.

Online co-regularization algorithms	Gradient ascent	ADA $S_t = \{t\}$	ADA Buffer- N	ADA Buffer- L
Error rates (%)				
No sparse approximation	12.75	7.75	6.43	20.35
Absolute threshold ($\varepsilon = 0.05$)	11.55	10.75	9.27	18.75
k -MC ($k = 400$)	11.55	10.35	9.45	18.35

FIGURE 8: Rotating two-moons-two-lines data sequence. We spin the two-moons-two-lines data set in top left of the two views during the sequence so that two moons and two lines smoothly rotate 360° in every 8000 examples.

7. Conclusion and Further Discussion

In this paper we presented an online coregularization framework based on the notion of ascending the dual function. We demonstrated that the existing online coregularization algorithms in previous work can be viewed as an approximation of our dual ascending process using gradient ascent. New online coregularization algorithms are derived based

on aggressive dual ascending procedures. For practical purpose, we proposed two sparse approximation approaches for kernel representation to reduce the computational complexity. Experiments showed that our online coregularization algorithms can adjust the boundary vector with the input sequence and have risk and error rates comparable to offline algorithms. Specially, our online coregularization algorithms can handle the settings where the target boundaries are

FIGURE 9: Error rates of ADA (Buffer-N) with different buffer sizes on the rotating two-moons-two-lines synthetic data sequence.

not fixed but rather drift with the sequence of examples in different views.

There are many interesting questions remaining in the online semi-supervised learning setting. For instance, we plan to study new online learning algorithms for other semi-supervised learning models. Another direction is how to choose effective combination of the parameter values more intelligently during the online coregularization process.

Appendix

Fenchel Conjugate

The Fenchel conjugate of a function $f: S \rightarrow \mathbb{R}$ is defined as

$$f^*(\lambda) = \sup \{ \langle \lambda, \omega \rangle - f(\omega) : \omega \in S \}. \quad (\text{A.1})$$

Since f^* is defined as a supremum of linear functions, it is convex. Here, we describe a few lemmas of Fenchel conjugate which we use as theoretical tools in this paper. More details are in [24].

Lemma A.1. Let f be a closed and convex function and let $\partial f(\omega)$ be its differential set at ω . Then, for all $\lambda \in \partial f(\omega)$, we have $f(\omega) + f^*(\lambda) = \langle \lambda, \omega \rangle$.

Proof. Since $\lambda \in \partial f(\omega)$ and f is closed, and convex, we know that $f(\omega') - f(\omega) \geq \langle \lambda, \omega' - \omega \rangle$ for all $\omega' \in S$. Equivalently

$$\langle \lambda, \omega \rangle - f(\omega) \geq \sup \{ \langle \lambda, \omega' \rangle - f(\omega') : \omega' \in S \}. \quad (\text{A.2})$$

The right-hand side of the previous equation equals to $f^*(\lambda)$, and thus

$$\langle \lambda, \omega \rangle - f(\omega) \geq f^*(\lambda) \longrightarrow \langle \lambda, \omega \rangle - f^*(\lambda) \geq f(\omega). \quad (\text{A.3})$$

The assumption that f is closed and convex implies that f is the Fenchel conjugate of f^* . Thus,

$$f(\omega) = \sup \{ \langle \lambda', \omega \rangle - f^*(\lambda') : \lambda' \in S \} \geq \langle \lambda, \omega \rangle - f^*(\lambda). \quad (\text{A.4})$$

Combining the two inequalities, we have

$$f(\omega) + f^*(\lambda) = \langle \lambda, \omega \rangle. \quad (\text{A.5})$$

□

Lemma A.2. Let $f(\omega) = \sum_{i=1}^k [b_i - \langle \omega, \mathbf{x}_i \rangle]_+$, where for all $i \in \{1, 2, \dots, k\}$, $b_i \in \mathbb{R}$ and $\mathbf{x}_i \in \mathbb{R}^n$. The Fenchel conjugate of $f(\omega)$ is

$$f^*(\lambda) = \begin{cases} -\sum_{i=1}^k \alpha_i b_i & \text{if } \lambda \in \left\{ -\sum_{i=1}^k \alpha_i \mathbf{x}_i : \forall i \in \{1, 2, \dots, k\}, \alpha_i \in [0, 1] \right\} \\ \infty & \text{otherwise.} \end{cases} \quad (\text{A.6})$$

Proof. We first rewrite the $f(\omega)$ as follows:

$$f(\omega) = \sum_{i=1}^k [b_i - \langle \omega, \mathbf{x}_i \rangle]_+ \quad (\text{A.7})$$

$$= \max_{\alpha_1, \alpha_2, \dots, \alpha_k \in [0, 1]} \sum_{i=1}^k \alpha_i (b_i - \langle \omega, \mathbf{x}_i \rangle),$$

where $\alpha_i \in [0, 1]$ for all $i \in \{1, 2, \dots, k\}$. Based on the definition of Fenchel conjugate, we can obtain that

$$\begin{aligned} f^*(\lambda) &= \max_{\omega} (\langle \lambda, \omega \rangle - f(\omega)) \\ &= \max_{\omega} \left(\langle \lambda, \omega \rangle - \max_{\alpha_1, \alpha_2, \dots, \alpha_k \in [0, 1]} \sum_{i=1}^k \alpha_i (b_i - \langle \omega, \mathbf{x}_i \rangle) \right) \\ &= \max_{\omega} \min_{\alpha_1, \alpha_2, \dots, \alpha_k \in [0, 1]} \left(\langle \lambda, \omega \rangle - \sum_{i=1}^k \alpha_i (b_i - \langle \omega, \mathbf{x}_i \rangle) \right) \\ &= \min_{\alpha_1, \alpha_2, \dots, \alpha_k \in [0, 1]} \max_{\omega} \left(-\sum_{i=1}^k \alpha_i b_i + \left\langle \lambda + \sum_{i=1}^k \alpha_i \mathbf{x}_i, \omega \right\rangle \right) \\ &= \min_{\alpha_1, \alpha_2, \dots, \alpha_k \in [0, 1]} \left(-\sum_{i=1}^k \alpha_i b_i + \max_{\omega} \left\langle \lambda + \sum_{i=1}^k \alpha_i \mathbf{x}_i, \omega \right\rangle \right). \end{aligned} \quad (\text{A.8})$$

Since the aforementioned third equality follows from the strong max-min property, it can be transferred into a min-max problem. If $\lambda + \sum_{i=1}^k \alpha_i \mathbf{x}_i \neq 0$, $\max_{\omega} \langle \lambda + \sum_{i=1}^k \alpha_i \mathbf{x}_i, \omega \rangle$ is ∞ ; otherwise, if $\lambda + \sum_{i=1}^k \alpha_i \mathbf{x}_i = 0$, we have $f^*(\lambda) = -\sum_{i=1}^k \alpha_i b_i$. □

Lemma A.3. Let $\|\cdot\|$ be any norm on $\mathbb{R}^n$, and let $f(\omega) = (1/2)\|\omega\|^2$ with $S = \mathbb{R}^n$. Then $f^*(\lambda) = (1/2)\|\lambda\|_*^2$ where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$. The domain of f^* is also $\mathbb{R}^n$. For example, if $f(\omega) = (1/2)\|\omega\|_2^2$ then $f^*(\lambda) = (1/2)\|\lambda\|_2^2$ since ℓ_2 norm is dual to itself.

Lemma A.4. Let f be a function, and let f^* be its Fenchel conjugate. For $a > 0$ and $b \in \mathbb{R}$, the Fenchel conjugate of $g(\omega) = af(\omega) + b$ is $g^*(\lambda) = af^*(\lambda/a) - b$.

References

- [1] Y. Altun, D. McAllester, and M. Belkin, "Maximum margin semi-supervised learning for structured variables," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 18, 2005.

- [2] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT '98)*, pp. 92–100, July 1998.
- [3] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.
- [4] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Large scale transductive SVMs," *Journal of Machine Learning Research*, vol. 7, pp. 1687–1712, 2006.
- [5] X. Zhu, "Semi-supervised learning literature survey," Tech. Rep. 1530, University of Wisconsin Madison, 2008.
- [6] Y. Peng, G. Kou, Y. Shi, and Z. Chen, "A descriptive framework for the field of data mining and knowledge discovery," *International Journal of Information Technology and Decision Making*, vol. 7, no. 4, pp. 639–682, 2008.
- [7] A. Blum, "On-line algorithms in machine learning," in *Online Algorithms: The State of the Art*, A. Fiat and G. Woeginger, Eds., vol. 1442 of *Lecture Notes in Computer Science*, 1998.
- [8] P. Joshi and P. Kulkarni, "Incremental Learning: areas and methods—a survey," *International Journal of Data Mining & Knowledge Management Process*, vol. 2, no. 5, 2012.
- [9] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.
- [10] T. de Ruijter and E. Tsivtsivadze, "Online co-regularized algorithms," in *Discovery Science*, vol. 7569 of *Lecture Notes in Computer Science*, pp. 184–193, Springer, 2012.
- [11] A. B. Goldberg, M. Li, and X. Zhu, "Online manifold regularization: a new learning setting and empirical study," in *Machine Learning and Knowledge Discovery in Databases*, vol. 5211 of *Lecture Notes in Computer Science*, pp. 393–407, 2008.
- [12] M. Bilenko and R. J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pp. 39–48, Washington, DC, USA, August 2003.
- [13] D. Cohn, R. Caruana, and A. McCallum, "Semi-supervised clustering with user feedback," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2000.
- [14] D. Klein, S. D. Kamvar, and C. D. Manning, "From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering," in *Proceedings of the 19th International Conference on Machine Learning*, pp. 307–314, Morgan Kaufmann Publishers, 2002.
- [15] N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and semi-supervised clustering: a brief survey," in *A Review of Machine Learning Techniques for Processing Multimedia Content, Report of the MUSCLE European Network of Excellence (FP6)*, 2004.
- [16] V. Sindhwani, P. Niyogi, and M. Belkin, "A co-regularized approach to semi-supervised learning with multiple views," in *Proceedings of the 22nd ICML Workshop on Learning with Multiple Views*, August 2005.
- [17] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 144–152, July 1992.
- [18] S. Szedmak and J. Shawe-Taylor, "Synthesis of maximum margin and multiview learning using unlabeled data," *Neurocomputing*, vol. 70, no. 7–9, pp. 1254–1264, 2007.
- [19] S. Sun and J. Shawe-Taylor, "Sparse semi-supervised learning using conjugate functions," *Journal of Machine Learning Research*, vol. 11, pp. 2423–2455, 2010.
- [20] B. L. Sun, G. H. Li, L. Jia, and H. Zhang, "Online manifold regularization by dual ascending procedure," *Mathematical Problems in Engineering*, vol. 2013, Article ID 838439, 16 pages, 2013.
- [21] A. Levy and M. Lindenbaum, "Sequential Karhunen-Loeve basis extraction and its application to images," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1371–1374, 2000.
- [22] M. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, pp. 130–137, 1980.
- [23] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, 1988.
- [24] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 2nd edition, 1995.