Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 838439, 16 pages
http://dx.doi.org/10.1155/2013/838439

Research Article

Online Manifold Regularization by Dual Ascending Procedure

Boliang Sun, Guohui Li, Li Jia, and Hui Zhang

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

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

Received 25 March 2013; Accepted 6 June 2013

Academic Editor: Zheng-Guang Wu

We propose a novel online manifold regularization framework based on the notion of duality in constrained optimization. The Fenchel conjugate of hinge functions is a key to transfer manifold regularization from offline to online in this paper. Our algorithms are derived by gradient ascent in the dual function. For practical purpose, we propose two buffering strategies and two sparse approximations to reduce the computational complexity. Detailed experiments verify the utility of our approaches. An important conclusion is that our online MR algorithms can handle the settings where the target hypothesis is not fixed but drifts with the sequence of examples. We also recap and draw connections to earlier works. This paper paves a way to the design and analysis of online manifold regularization algorithms.

1. Introduction

Semisupervised learning (S^2L) of different classifiers is an important problem in machine learning with interesting theoretical properties and practical applications [1–5]. Different from standard supervised learning (SL), the S^2L paradigm learns from both labeled and unlabeled examples. In this paper, we investigate the online semisupervised learning (OS^2L) problems which have three features as follows:

(i) data is abundant but the resources to label them are limited;

(ii) data arrives in a stream and cannot even store them all;

(iii) no statistical assumptions are found, which means that $p(\mathbf{x}, y)$ can change over time.

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. A popular idea [5, 6] is using a heuristic method to greedily label the unlabeled examples, which is essentially still employing an online supervised learning framework. References [7–9] also treat OS^2L problem as online semisupervised 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 [10]).

For solving OS^2L problem, we introduce a novel online manifold regularization (MR) framework for the design and analysis of new online MR algorithms in this paper. Manifold regularization is a geometric framework for learning from examples. This idea of regularization exploits the geometry of the probability distribution that generates the data and incorporates it as an additional regularization term. Hence, the objective function has two regularization terms: one controls the complexity of the classifier in the ambient space and the other controls the complexity as measured by the geometry of the distribution.

Since decreasing the primal MR objective function is impossible before obtaining all the training examples, we propose a Fenchel conjugate transform to optimize the dual

problem in an online manner. Unfortunately, the basic online MR algorithms derived from our framework have to store all the incoming examples and the time complexity on each learning round is $O(t^2)$. Therefore, we propose two buffering strategies and two sparse approximations to make our online MR algorithms practical. 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 an empirical online version of manifold regularization of SVMs [11]. Their method defines an instantaneous regularized risk to avoid optimizing the primal MR problem directly. The learning process is based on convex programming with stochastic gradient descent in kernel space. The update scheme of this work can be derived from our online MR framework.

This paper is structured as follows. In Section 2 we begin with a primal view of semisupervised learning problem based on manifold regularization. In Section 3, our new framework for designing and analyzing online MR algorithms is introduced. Next, in Section 4, we derive new algorithms from our online MR framework by gradient ascent. In Section 5, we propose two sparse approximations for kernel representation to reduce computational complexity. Connections to earlier analysis techniques are in Section 6. Experiments and analyses are in Section 7. In Section 8, possible extensions of our work are given.

2. 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., $\boldsymbol{\omega}$ and $\boldsymbol{\lambda}$). $(\mathbf{x}_t, y_t, \sigma_t)$ denotes the tth training example, where $\mathbf{x}_t \in \mathbb{R}^n$ is the point, 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 $[a]_+ = \max\{a, 0\}$. $\langle \boldsymbol{\omega}, \mathbf{x} \rangle$ denotes the inner product between vectors $\boldsymbol{\omega}$ and $\mathbf{x}$. For any $t \geq 1$, the set of integers $\{1, 2, \ldots, t\}$ is denoted by $[t]$.

Consider an input sequence $(\mathbf{x}_1, y_1, \sigma_1), (\mathbf{x}_2, y_2, \sigma_2), \ldots, (\mathbf{x}_T, y_T, \sigma_T)$, where $\mathbf{x}_t \in \mathbb{R}^n$ and $\sigma_t \in \{0, 1\}$ $(t \in \{1, 2, \ldots, T\})$. Let K be a kernel over the training points $\mathbf{x}$ and $\mathscr{H}_K$ the corresponding reproducing kernel Hilbert space (RKHS). The S^{2}L problem based on manifold regularization [12] can be written as minimizing

$$J(f) = \frac{1}{2}\|f\|_K^2 + c_1 \sum_{t=1}^{T} \sigma_t h\left(f\left(\mathbf{x}_t\right), y_t\right) + c_2 \sum_{i,j=1}^{T} w_{ij} d\left(f\left(\mathbf{x}_i\right), f\left(\mathbf{x}_j\right)\right), \tag{1}$$

where $f \in \mathscr{H}_K$, $\|f\|_K^2$ is the RKHS norm of f, h is a loss function for the predictions of the training points, c_1 and c_2 are trade-off parameters, $d(f(\mathbf{x}_i), f(\mathbf{x}_j))$ is the distance function which measures the difference between the predictions of $\mathbf{x}_i$ and $\mathbf{x}_j$, and w_{ij} are the edge weights which define a graph over the T examples, for example, a fully connected graph with Gaussian weights $w_{ij} = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2}$ or k-NN binary weights.

In (1), the objective function $J(f)$ can be composed of three sums. The first sum measures the complexity of f, the second measures the loss for labeled examples, and the last one is the manifold regularizer which encourages prediction smoothness over the graph which means that similar examples tend to have same predictions.

Denote that $f^* = \operatorname{argmin}_{f \in \mathscr{H}_K} J(f)$. Obviously, it is easy to seek f^* using existing optimization tools after all the training examples arrived, which is called offline MR. Different from offline methods, an online MR process is performed in sequence of consecutive rounds. On each round, when an example $(\mathbf{x}, y, \sigma)$ arrives, the online MR algorithm is required to present its predictive label and update its prediction mechanism so as to be more accurate later.

For simplicity and concreteness, we focus on semisupervised binary linear classifiers in this paper, which means that $f(\mathbf{x}) = \langle \boldsymbol{\omega}, \mathbf{x} \rangle$ and the data labels belong to $\{-1, +1\}$. h is chosen as a popular convex loss function in supervised classification: hinge-loss, defined as

$$h\left(f\left(\mathbf{x}_t\right), y_t\right) = \left[1 - y_t f\left(\mathbf{x}_t\right)\right]_+ = \left[1 - y_t \left\langle \boldsymbol{\omega}, \mathbf{x}_t \right\rangle\right]_+. \tag{2}$$

The function $d(f(\mathbf{x}_i), f(\mathbf{x}_j))$ is defined as an absolute function in this paper, where

$$d\left(f\left(\mathbf{x}_i\right), f\left(\mathbf{x}_j\right)\right) = \left|f\left(\mathbf{x}_i\right) - f\left(\mathbf{x}_j\right)\right|. \tag{3}$$

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

$$d\left(f\left(\mathbf{x}_i\right), f\left(\mathbf{x}_j\right)\right) = \left[f\left(\mathbf{x}_i\right) - f\left(\mathbf{x}_j\right)\right]_+ + \left[f\left(\mathbf{x}_j\right) - f\left(\mathbf{x}_i\right)\right]_+. \tag{4}$$

To learn a max-margin decision boundary, we can rewrite (1) as

$$J(\boldsymbol{\omega}) = \frac{1}{2}\boldsymbol{\omega}^2 + c_1 \sum_{t=1}^{T} \sigma_t \left[1 - y_t \left\langle \boldsymbol{\omega}, \mathbf{x}_t \right\rangle\right]_+ + c_2 \sum_{i,j=1}^{T} \left|w_{ij} \left\langle \boldsymbol{\omega}, \mathbf{x}_i - \mathbf{x}_j \right\rangle\right|. \tag{5}$$

Let edge weights $w_{ij} = w_{ji}$ and $g_t(\boldsymbol{\omega}) = c_1 \sigma_t [1 - y_t \langle \boldsymbol{\omega}, \mathbf{x}_t \rangle]_+ + 2c_2 \sum_{i=1}^{t-1} |w_{ij} \langle \boldsymbol{\omega}, \mathbf{x}_i - \mathbf{x}_j \rangle|$, and we can get a simple version of (5), as

$$J(\boldsymbol{\omega}) = \frac{1}{2}\boldsymbol{\omega}^2 + \sum_{t=1}^{T} g_t(\boldsymbol{\omega}). \tag{6}$$

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

3. Online Manifold Regularization in the Dual Problem

In this section, we propose a unified online manifold regularization framework of semisupervised binary classification

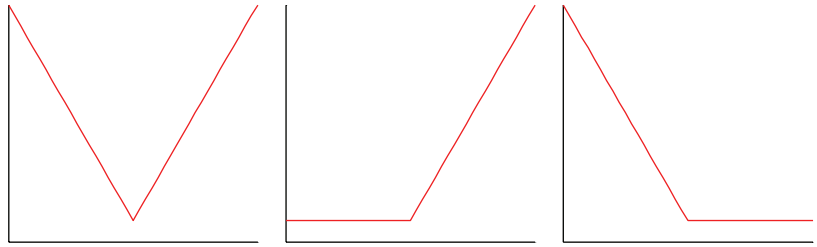

Figure 1: The absolute distance function and its components. The absolute function $|\mathbf{x}|$ (a) can be decomposed into the sum of two hinge functions $[\mathbf{x}]_+$ (b) and $[-\mathbf{x}]_+$ (c).

problems. Our presentation reveals how the S^{2}L problem based on MR 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. The Fenchel conjugate of a function $f : \text{dom } f \to \mathbb{R}$ is defined as

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

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

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

$$f^*(\boldsymbol{\lambda}) = \begin{cases} -\sum_{i=1}^{k} \alpha_i b_i & \text{if } \boldsymbol{\lambda} \in \left\{-\sum_{i=1}^{k} \alpha_i \mathbf{x}_i : \forall i \in \{1, 2, \ldots, k\},\ \alpha_i \in [0, 1]\right\} \\ \infty & \text{otherwise.} \end{cases} \tag{8}$$

Proof. We first rewrite the $f(\boldsymbol{\omega})$ as the following:

$$\begin{aligned} f(\boldsymbol{\omega}) &= \sum_{i=1}^{k} [b_i - \langle \boldsymbol{\omega}, \mathbf{x}_i\rangle]_+ \\ &= \max_{\alpha_1, \alpha_2, \ldots, \alpha_k \in [0,1]} \sum_{i=1}^{k} \alpha_i (b_i - \langle \boldsymbol{\omega}, \mathbf{x}_i\rangle), \end{aligned} \tag{9}$$

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

$$\begin{aligned} f^*(\boldsymbol{\lambda}) &= \max_{\boldsymbol{\omega}} (\langle \boldsymbol{\lambda}, \boldsymbol{\omega}\rangle - f(\boldsymbol{\omega})) \\ &= \max_{\boldsymbol{\omega}} \left(\langle \boldsymbol{\lambda}, \boldsymbol{\omega}\rangle - \max_{\alpha_1, \alpha_2, \ldots, \alpha_k \in [0,1]} \sum_{i=1}^{k} \alpha_i (b_i - \langle \boldsymbol{\omega}, \mathbf{x}_i\rangle) \right) \\ &= \max_{\boldsymbol{\omega}} \min_{\alpha_1, \alpha_2, \ldots, \alpha_k \in [0,1]} \left(\langle \boldsymbol{\lambda}, \boldsymbol{\omega}\rangle - \sum_{i=1}^{k} \alpha_i (b_i - \langle \boldsymbol{\omega}, \mathbf{x}_i\rangle) \right) \\ &= \min_{\alpha_1, \alpha_2, \ldots, \alpha_k \in [0,1]} \max_{\boldsymbol{\omega}} \left(-\sum_{i=1}^{k} \alpha_i b_i + \left\langle \boldsymbol{\lambda} + \sum_{i=1}^{k} \alpha_i \mathbf{x}_i, \boldsymbol{\omega} \right\rangle \right) \\ &= \min_{\alpha_1, \alpha_2, \ldots, \alpha_k \in [0,1]} \left(-\sum_{i=1}^{k} \alpha_i b_i + \max_{\boldsymbol{\omega}} \left\langle \boldsymbol{\lambda} + \sum_{i=1}^{k} \alpha_i \mathbf{x}_i, \boldsymbol{\omega} \right\rangle \right). \end{aligned} \tag{10}$$

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

Back to the primal problem, we want to get a sequence of boundary $\boldsymbol{\omega}_0, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_T$ which makes $J(\boldsymbol{\omega}_0) \geq J(\boldsymbol{\omega}_1) \geq \cdots \geq J(\boldsymbol{\omega}_T)$. In (6), decreasing the objective function $J(\boldsymbol{\omega})$ directly is impossible in the condition of not getting all the training examples. In practice, we only get the example set $\{(\mathbf{x}_1, y_1, \sigma_1), (\mathbf{x}_2, y_2, \sigma_2), \ldots, (\mathbf{x}_t, y_t, \sigma_t)\}$ on round t, when the training examples arrive in a steam. In order to avoid the previous contradiction, we propose a Fenchel conjugate transform of S^{2}L problems based on MR.

An equivalent problem of (6) is

$$\begin{aligned} &\min_{\boldsymbol{\omega}_0, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_T} \frac{1}{2}\boldsymbol{\omega}_0^2 + \sum_{t=1}^{T} g_t(\boldsymbol{\omega}_t), \\ &\text{s.t.} \qquad \forall i \in 1, 2 \ldots, T,\ \boldsymbol{\omega}_i = \boldsymbol{\omega}_0. \end{aligned} \tag{11}$$

Using the Lagrange dual function, we can rewrite (11) by introducing a vector group $(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_T)$:

$$\max_{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_T} \min_{\boldsymbol{\omega}_0, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_T} \frac{1}{2}\boldsymbol{\omega}_0^2 + \sum_{t=1}^{T} g_t(\boldsymbol{\omega}_t) + \sum_{t=1}^{T} \langle \boldsymbol{\lambda}_t, \boldsymbol{\omega}_0 - \boldsymbol{\omega}_t\rangle. \tag{12}$$

Consider the dual function

$$D(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_T) = \min_{\boldsymbol{\omega}_0, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_T} \frac{1}{2}\boldsymbol{\omega}_0^2 + \sum_{t=1}^{T} g_t(\boldsymbol{\omega}_t) + \sum_{t=1}^{T} \langle \boldsymbol{\lambda}_i, \boldsymbol{\omega}_0 - \boldsymbol{\omega}_t\rangle$$

$$= -\max_{\boldsymbol{\omega}_0}\left(\sum_{t=1}^{T}\langle -\boldsymbol{\lambda}_t, \boldsymbol{\omega}_0\rangle - \frac{1}{2}\boldsymbol{\omega}_0^2\right) - \sum_{t=1}^{T}\max_{\boldsymbol{\omega}_t}\left(\langle \boldsymbol{\lambda}_t, \boldsymbol{\omega}_t\rangle - g_t(\boldsymbol{\omega}_t)\right) = -\frac{1}{2}\left(-\sum_{t=1}^{T}\boldsymbol{\lambda}_t\right)^2 - \sum_{t=1}^{T} g_t^*(\boldsymbol{\lambda}_t), \tag{13}$$

where g_t^* is the Fenchel conjugate of g_t. The primal problem can be described by Fenchel conjugate transform as follows:

$$\min_{\boldsymbol{\omega}}\frac{1}{2}\boldsymbol{\omega}^2 + \sum_{t=1}^{T} g_t(\boldsymbol{\omega}) = \max_{\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2,\ldots,\boldsymbol{\lambda}_T} D(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_T) = \max_{\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2,\ldots,\boldsymbol{\lambda}_T} -\frac{1}{2}\left(-\sum_{t=1}^{T}\boldsymbol{\lambda}_t\right)^2 - \sum_{t=1}^{T} g_t^*(\boldsymbol{\lambda}_t). \tag{14}$$

In (14), we can see that our goal has been transferred from minimizing the primal problem $J(\boldsymbol{\omega})$ to maximizing the dual function $D(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_T)$. In the following, we show how to ascend the dual function without the unobserved examples.

Based on Proposition 1, the Fenchel conjugate of $g_t(\boldsymbol{\omega})$ is

$$g_t^*(\boldsymbol{\lambda}) = -c_1\alpha_{t0}\sigma_t \tag{15}$$

if $\boldsymbol{\lambda}_t \in \{-c_1\alpha_{t0}\sigma_t y_t \mathbf{x}_t - \sum_{i=1}^{t-1} 2c_2(\alpha_{ti}^1 - \alpha_{ti}^2)w_{ti}(\mathbf{x}_t - \mathbf{x}_i),\ \alpha_{t0} \in [0,1]$ and for all $i \in \{1, 2, \ldots, t-1\}$, $\alpha_{ti}^1, \alpha_{ti}^2 \in [0,1]\}$; otherwise, $g_t^*(\boldsymbol{\lambda}_t) = \infty$.

Since our goal is to maximize the dual function, we can restrict to the case that $\boldsymbol{\lambda}_t \in \{-c_1\alpha_{t0}\sigma_t y_t \mathbf{x}_t - \sum_{i=1}^{t-1} 2c_2(\alpha_{ti}^1 - \alpha_{ti}^2)w_{ti}(\mathbf{x}_t - \mathbf{x}_i),\ \alpha_{t0} \in [0,1]$ and for all $i \in \{1, 2, \ldots, t-1\}$, $\alpha_{ti}^1, \alpha_{ti}^2 \in [0,1]\}$, where $t \in \{1, 2, \ldots, T\}$. $g_t^*(\boldsymbol{\lambda}_t)$ has $2t-1$ associated coefficients which are $\alpha_{t0}, \alpha_{t1}^1, \alpha_{t1}^2, \ldots, \alpha_{t(t-1)}^1, \alpha_{t(t-1)}^2$.

Based on the previous analysis, the dual function can be rewritten using a new coefficient vector $\boldsymbol{\alpha} = [\alpha_{10}, \alpha_{20}, \alpha_{21}^1, \alpha_{21}^2, \ldots, \alpha_{T0}, \alpha_{T1}^1, \alpha_{T1}^2, \ldots, \alpha_{T(T-1)}^1, \alpha_{T(T-1)}^2]$,

$$\begin{aligned} D(\boldsymbol{\alpha}) &= D\left(\alpha_{10}, \alpha_{20}, \alpha_{21}^1, \alpha_{21}^2, \ldots, \alpha_{T0}, \alpha_{T1}^1, \alpha_{T1}^2, \ldots, \alpha_{T(T-1)}^1, \alpha_{T(T-1)}^2\right) \\ &= -\frac{1}{2}\left(\sum_{t=1}^{T}\left(c_1\alpha_{t0}\sigma_t y_t \mathbf{x}_t + \sum_{i=1}^{t-1} 2c_2\left(\alpha_{ti}^1 - \alpha_{ti}^2\right)w_{ti}\left(\mathbf{x}_t - \mathbf{x}_i\right)\right)\right)^2 + \sum_{t=1}^{T} c_1\alpha_{t0}\sigma_t. \end{aligned} \tag{16}$$

And our online MR task can be redescribed as ascending the dual function $D(\boldsymbol{\alpha})$ by updating the coefficient vector $\boldsymbol{\alpha}$. Obviously, unobserved examples would make no influence on the value of dual function in (16) by setting their associate coefficients to zero.

Denote $\boldsymbol{\alpha}_t$ to be the coefficient vector $\boldsymbol{\alpha}$ on round t, and its elements can be written as $(\alpha_{10})_t, (\alpha_{20})_t, (\alpha_{21}^1)_t, (\alpha_{21}^2)_t, \ldots, (\alpha_{T0})_t, (\alpha_{T1}^1)_t, (\alpha_{T1}^2)_t, \ldots, (\alpha_{T(T-1)}^1)_t, (\alpha_{T(T-1)}^2)_t$. The update process of coefficient vector $\boldsymbol{\alpha}$ on round t should satisfy the following conditions:

(i) If $t+1 \le i \le T$, $(\alpha_{i0})_t, (\alpha_{i1}^1)_t, (\alpha_{i1}^2)_t, \ldots, (\alpha_{i(i-1)}^1)_t, (\alpha_{i(i-1)}^2)_t = 0$;

(ii) $D(\boldsymbol{\alpha}_t) \ge D(\boldsymbol{\alpha}_{t-1})$.

The first one means that the unobserved examples do not make influence on the value of dual function $D(\boldsymbol{\alpha}_t)$, and the second means that the value of dual function never decreases along the online MR process. Therefore, the dual function on round t can be written as

$$D(\boldsymbol{\alpha}_t) = -\frac{1}{2}\left(\sum_{i=1}^{t}\left(c_1(\alpha_{i0})_t\sigma_i y_i \mathbf{x}_i + \sum_{j=1}^{i-1} 2c_2\left(\left(\alpha_{ij}^1\right)_t - \left(\alpha_{ij}^2\right)_t\right)w_{ij}\left(\mathbf{x}_i - \mathbf{x}_j\right)\right)\right)^2 + \sum_{i=1}^{t} c_1(\alpha_{i0})_t\sigma_i. \tag{17}$$

Based on Lemmas 2 and 3 in the appendix, we can obtain that each coefficient vector $\boldsymbol{\alpha}$ has an associated boundary vector $\boldsymbol{\omega}$. On round t, the associated boundary vector of $\boldsymbol{\alpha}_t$ is

$$\begin{aligned} \boldsymbol{\omega}_t &= \sum_{i=1}^{t}\left(c_1(\alpha_{i0})_t\sigma_i y_i \mathbf{x}_i + \sum_{j=1}^{i-1} 2c_2\left(\left(\alpha_{ij}^1\right)_t - \left(\alpha_{ij}^2\right)_t\right)w_{ij}\left(\mathbf{x}_i - \mathbf{x}_j\right)\right) \\ &= \sum_{i=1}^{t}\left(c_1(\alpha_{i0})_t\sigma_i y_i + \sum_{j=1}^{i-1} 2c_2\left(\left(\alpha_{ij}^1\right)_t - \left(\alpha_{ij}^2\right)_t\right)w_{ij} - \sum_{j=i+1}^{t} 2c_2\left(\left(\alpha_{ji}^1\right)_t - \left(\alpha_{ji}^2\right)_t\right)w_{ji}\right)\mathbf{x}_i. \end{aligned} \tag{18}$$

Using a more general form, the associate vector $\boldsymbol{\omega}_t$ in (18) also can be written as

$$\boldsymbol{\omega}_t = \sum_{i=1}^{t}(\beta_i)_t\mathbf{x}_i, \tag{19}$$

where $(\beta_i)_t = c_1(\alpha_{i0})_t\sigma_i y_i + \sum_{j=1}^{i-1} 2c_2((\alpha_{ij}^1)_t - (\alpha_{ij}^2)_t)w_{ij} - \sum_{j=i+1}^{t} 2c_2((\alpha_{ji}^1)_t - (\alpha_{ji}^2)_t)w_{ji}$.

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

INPUT: two positive scalars: c_1 and c_2; edge weights w_{ij}.
INITIALIZE: a coefficient vector $\boldsymbol{\alpha}_0$ and its associated decision boundary vector $\boldsymbol{\omega}_0$.
PROCESS: For $t = 1, 2, \ldots, T$
Receive an example $(\mathbf{x}_t, y_t, \sigma_t)$,
Choose a new coefficient vector $\boldsymbol{\alpha}_t$ that satisfies $D(\boldsymbol{\alpha}_t) \geq D(\boldsymbol{\alpha}_{t-1})$,
Return a new associated boundary vector $\boldsymbol{\omega}_t$ in (18),
If $\sigma_t = 0$, predict $\widehat{y}_t = \text{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle)$.

Algorithm 1: A template online manifold regularization algorithm for semi-supervised binary classification. Based on dual ascending procedure, this template algorithm aims for an increment of the dual function on each round.

4. Deriving New Algorithms by Gradient Ascent

In the previous section, a template algorithm framework for online MR is proposed based on the idea of ascending the dual function. In this section we derive different online MR algorithms using different update schemes of coefficient vector $\boldsymbol{\alpha}$ in the dual function.

Let I_t denote a subset of dual coefficients and α is an element of coefficient vector $\boldsymbol{\alpha}$. Our online MR algorithms simply perform a gradient ascent step over I_t ($t \in \{1, 2, \ldots, T\}$) on round t that aims to increase the value of dual function:

$$(\alpha)_t = (\alpha)_{t-1} + \rho_t \frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial (\alpha)_{t-1}}, \tag{20}$$

where $\alpha \in I_t$ and $\rho_t \geq 0$ is a step size. We now propose three update schemes which modify different coefficients on each learning round.

4.1. Example-Associate (EA) Update. In traditional online supervised learning, the prediction mechanism is always updated only using the new arrived example, for example, Perceptron. Based on this notion, we propose an example-associate update scheme to ascend the dual function by updating the associated coefficients of the new training example $(\mathbf{x}_t, y_t, \sigma_t)$ on round t that means $I_t \in \{\alpha_{t0}, \alpha_{t1}^1, \alpha_{t1}^2, \ldots, \alpha_{t(t-1)}^1, \alpha_{t(t-1)}^2\}$.

In online MR process, the coefficients $\alpha_{t0}, \alpha_{t1}^1, \alpha_{t1}^2, \ldots, \alpha_{t(t-1)}^1, \alpha_{t(t-1)}^2$ do not need to be grounded to zero on round t. Based on Proposition 1, we have already obtained that every element of coefficient vector $\boldsymbol{\alpha}$ belongs to $[0, 1]$. Using a gradient ascent step in (20), the example-associate update process can be written as

$$(\alpha_{t0})_t = \rho_t c_1 \sigma_t [1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \rangle]_+, \tag{21}$$

$$\left(\alpha_{ti}^1\right)_t = 2c_2 \rho_t [-w_{ti} \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle]_+, \quad i \in \{1, 2, \ldots, t-1\}, \tag{22}$$

$$\left(\alpha_{ti}^2\right)_t = 2c_2 \rho_t [w_{ti} \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle]_+, \quad i \in \{1, 2, \ldots, t-1\}. \tag{23}$$

Equation (21) implies that if $1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \rangle < 0$, we have $(\alpha_{t0})_t = (\alpha_{t0})_{t-1} = 0$, and otherwise $(\alpha_{t0})_t = \rho_t c_1 \sigma_t (1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \rangle)$. Equations (22) and (23) also imply that the gradient ascent must satisfy $(\alpha)_t \geq 0$, and otherwise we do not perform a gradient ascent on α.

Unfortunately, this update scheme will not work in practice because it needs to store every input point to update the boundary vector; it also has an increasing time complexity $O(t)$. Here, we propose two buffering strategies to use a small buffer of examples on each learning round. Denote that $S_t \subseteq [t-1]$, and the example $(\mathbf{x}_i, y_i, \sigma_i)$ belongs to the buffer on round t if $i \in S_t$.

(i) *Buffer-N*. Let the buffer size be τ. τ-buffer replaces the oldest point $\mathbf{x}_{t-\tau}$ in the buffer with the new incoming point $\mathbf{x}_t$ after each learning round, which means that $S_t = \{t - \tau, t - \tau + 1, \ldots, t - 1\}$.

(ii) *Buffer-U*. 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.

Based on the previous analysis, the sub set of dual coefficients I_t can be chosen using the process in Algorithm 2.

Denote $\rho_t^{\max}$ as the maximal step size on round t. Since every dual coefficient belongs to $[0, 1]$, we have $\rho_t^{\max} = \min\{1/c_1\sigma_t[1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \rangle]_+, \ 1/2c_2[-w_{ti}\langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle]_+, 1/2c_2[w_{ti}\langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle]_+\}$, $i \in S_t$. The optimal step size ρ_t^* is

$$\rho_t^* = \frac{\langle \partial D(\boldsymbol{\alpha}_{t-1}) / \partial I_t, \partial D(\boldsymbol{\alpha}_{t-1}) / \partial I_t \rangle}{\langle \partial D(\boldsymbol{\alpha}_{t-1}) / \partial I_t, H(\boldsymbol{\alpha}_{t-1}) (\partial D(\boldsymbol{\alpha}_{t-1}) / \partial I_t) \rangle}, \tag{24}$$

where $H(\boldsymbol{\alpha}_{t-1})$ is the Hessian of $D(\boldsymbol{\alpha}_{t-1})$ over I_t. Then, we obtain that if $\rho_t \in [0, \min\{\rho_t^{\max}, \rho_t^*\}]$, $D(\boldsymbol{\alpha}_t) \geq D(\boldsymbol{\alpha}_{t-1})$.

Combining (22) and (23), we have

$$\begin{aligned} &\left(\alpha_{ti}^1\right)_t - \left(\alpha_{ti}^2\right)_t \\ &\quad = 2c_2\rho_t \left([-w_{ti} \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle]_+ \right. \\ &\qquad \left. - [w_{ti} \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle]_+ \right) \\ &\quad = -2c_2\rho_t w_{ti} \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle. \end{aligned} \tag{25}$$

PROCESS: $I_t = \emptyset$,
If $1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \rangle > 0$, $I_t = I_t \cup \{\alpha_{t0}\}$.
For each $i \in S_t$
If $-w_{ti} \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle > 0$, $I_t = I_t \cup \{\alpha_{ti}^1\}$;
If $w_{ti} \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle > 0$, $I_t = I_t \cup \{\alpha_{ti}^2\}$.
Return I_t.

Algorithm 2: The process of getting I_t for example-associate update.

We also can rewrite the update process using the form of (19) as follows:

$$\begin{aligned}(\beta_t)_t &= \rho_t c_1^2 \sigma_t y_t \left[1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \rangle\right]_+ \\ &\quad - 4\rho_t c_2^2 \sum_{i \in S_t} w_{ti}^2 \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle, \\ (\beta_i)_t &= (\beta_i)_{t-1} + 4\rho_t c_2^2 w_{ti}^2 \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle, \quad i \in S_t.\end{aligned} \tag{26}$$

The new associate boundary vector is

$$\begin{aligned}\boldsymbol{\omega}_t &= \boldsymbol{\omega}_{t-1} + \rho_t c_1^2 \sigma_t y_t \left[1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \rangle\right]_+ \mathbf{x}_t \\ &\quad - 4\rho_t c_2^2 \sum_{i \in S_t} w_{ti}^2 \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle (\mathbf{x}_t - \mathbf{x}_i).\end{aligned} \tag{27}$$

Algorithm 3 shows an online MR algorithm based on EA update.

Specially, while choosing a small stationary ρ on each learning round, we must have $D(\boldsymbol{\alpha}_t) \geq D(\boldsymbol{\alpha}_{t-1})$. In this condition, the update process of boundary vector can be written as

$$\begin{aligned}\boldsymbol{\omega}_t &= \boldsymbol{\omega}_{t-1} + \rho c_1^2 \sigma_t y_t \left[1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \rangle\right]_+ \mathbf{x}_t \\ &\quad - 4\rho c_2^2 \sum_{i \in S_t} w_{ti}^2 \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle (\mathbf{x}_t - \mathbf{x}_i).\end{aligned} \tag{28}$$

This update scheme is called ρ-EA update.

For an aggressive gradient ascent, we can choose $\rho_t = \min\{\rho_t^{\max}, \rho_t^*\}$, which is called aggressive-EA update.

4.2. Overall Update. In EA update scheme, we employ an additive way to update the boundary vector by updating the associated coefficients of the new training example. In fact, all the associated coefficients of arrived examples can be updated on each learning round. Here, we propose another new update scheme which is called overall update.

Using EA update, the dual function $D(\boldsymbol{\alpha}_t)$ on round t can be rewritten as

$$\begin{aligned}D(\boldsymbol{\alpha}_t) = &-\frac{1}{2}\Bigg(\boldsymbol{\omega}_{t-1} + c_1(\alpha_{t0})_t \sigma_t y_t \mathbf{x}_t \\ &+ \sum_{i=1}^{t-1} 2c_2 \left(\left(\alpha_{ti}^1\right)_t - \left(\alpha_{ti}^2\right)_t\right) w_{ti} (\mathbf{x}_t - \mathbf{x}_i)\Bigg)^2 \\ &+ \sum_{i=1}^{t} c_1(\alpha_{i0})_t \sigma_i.\end{aligned} \tag{29}$$

In fact, the dual coefficients in $\boldsymbol{\omega}_{t-1}$ also can be updated in (28). Since $\boldsymbol{\omega}_{t-1}$ has $(t-1)^2$ dual coefficients, it is impossible to update them, respectively. We introduce a new variable η_t into (29), as

$$\begin{aligned}D(\boldsymbol{\alpha}_t) = &-\frac{1}{2}\Bigg((1 - \eta_t)\boldsymbol{\omega}_{t-1} + c_1(\alpha_{t0})_t \sigma_t y_t \mathbf{x}_t \\ &+ \sum_{i=1}^{t-1} 2c_2 \left(\left(\alpha_{ti}^1\right)_t - \left(\alpha_{ti}^2\right)_t\right) w_{ti} (\mathbf{x}_t - \mathbf{x}_i)\Bigg)^2 \\ &+ (1 - \eta_t) \sum_{i=1}^{t-1} c_1(\alpha_{i0})_{t-1} \sigma_i + c_1(\alpha_{t0})_t \sigma_t.\end{aligned} \tag{30}$$

From (30), we can get that a gradient ascent update on η_t actually means to multiply all the dual coefficients in $\boldsymbol{\omega}_{t-1}$ by $1 - \eta_t$. Since every dual coefficient in $\boldsymbol{\omega}_{t-1}$ belongs to $[0, 1]$, we constrain $\eta_t \in [0, 1]$. The initial value of η_t is zero. Using a gradient ascent on η_t, we obtain

$$\eta_t = \rho_t \left[\langle \boldsymbol{\omega}_{t-1}, \boldsymbol{\omega}_{t-1} \rangle - \sum_{i=1}^{t-1} c_1(\alpha_{i0})_{t-1} \sigma_i\right]_+. \tag{31}$$

Therefore, we choose $I_t \in \{\eta_t, \alpha_{t0}, \alpha_{t1}^1, \alpha_{t1}^2, \ldots, \alpha_{t(t-1)}^1, \alpha_{t(t-1)}^2\}$ on each learning round for overall update. A buffering strategy also can be used in overall update, $S_t \subseteq [t-1]$. Like EA update, we propose a process to get I_t for overall update in Algorithm 4.

The gradient ascent for $\{\alpha_{t0}, \alpha_{t1}^1, \alpha_{t1}^2, \ldots, \alpha_{t(t-1)}^1, \alpha_{t(t-1)}^2\}$ is same as the EA update which has been shown in (21), (22), and (23). Combined with the constraint of η_t, the maximal step size for overall update is $\rho_t^{\max} = \min\{1/[\langle \boldsymbol{\omega}_{t-1}, \boldsymbol{\omega}_{t-1} \rangle - \sum_{i=1}^{t-1} c_1(\alpha_{i0})_{t-1}\sigma_i]_+, 1/c_1\sigma_t[1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \rangle]_+, 1/2c_2[-w_{ti}\langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle]_+, 1/2c_2[w_{ti}\langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle]_+\}$, $i \in S_t$. The optimal step size ρ_t^* also can be obtained using (24). Obviously, if $\rho_t \in [0, \min\{\rho_t^{\max}, \rho_t^*\}]$, $D(\boldsymbol{\alpha}_t) \geq D(\boldsymbol{\alpha}_{t-1})$. Rewriting the overall update process using the form of (19), we have

$$\begin{aligned}(\beta_t)_t &= \rho_t c_1^2 \sigma_t y_t \left[1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \rangle\right]_+ \\ &\quad - 4\rho_t c_2^2 \sum_{i \in S_t} w_{ti}^2 \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle, \\ (\beta_i)_t &= (1 - \eta_t)(\beta_i)_{t-1} + 4\rho_t c_2^2 w_{ti}^2 \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle, \quad i \in S_t.\end{aligned} \tag{32}$$

INPUT: two positive scalars: c_1 and c_2; edge weights w_{ij}.
INITIALIZE: a coefficient vector $\boldsymbol{\alpha}_0$ and its associated decision boundary vector $\boldsymbol{\omega}_0$.
PROCESS: For $t = 1, 2, \ldots, T$
Receive an example $(\mathbf{x}_t, y_t, \sigma_t)$,
Get I_t using the process in Algorithm 2,
Choose a step size $\rho_t \in [0, \min\{\rho_t^{\max}, \rho_t^*\}]$,
Update the boundary vector using (27),
If $\sigma_t = 0$, predict $\widehat{y}_t = \text{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle)$,
Renew the buffer.

Algorithm 3: Online manifold regularization algorithm based on EA update.

PROCESS: $I_t = \emptyset$,
If $\langle \boldsymbol{\omega}_{t-1}, \boldsymbol{\omega}_{t-1} \rangle - \sum_{i=1}^{t-1} c_1 (\alpha_{i0})_{t-1} \sigma_i > 0$,
$I_t = I_t \cup \{\eta_t\}$.
If $1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \rangle > 0$, $I_t = I_t \cup \{\alpha_{t0}\}$.
For each $i \in S_t$
If $-w_{ti} \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle > 0$, $I_t = I_t \cup \{\alpha_{ti}^1\}$;
If $w_{ti} \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle > 0$, $I_t = I_t \cup \{\alpha_{ti}^2\}$.
Return I_t.

Algorithm 4: The process of getting I_t for overall update.

The new associate boundary vector is

$$\boldsymbol{\omega}_t = (1 - \eta_t) \boldsymbol{\omega}_{t-1} + \rho_t c_1^2 \sigma_t y_t \left[1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \rangle\right]_+ \mathbf{x}_t - 4\rho_t c_2^2 \sum_{i \in S_t} w_{ti}^2 \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle (\mathbf{x}_t - \mathbf{x}_i). \tag{33}$$

Algorithm 5 shows the online MR algorithm based on overall update.

Like EA update, we also can derive ρ-overall update and aggressive-overall update from the previous analysis.

4.3. Two-Step Update. In the two update schemes aforementioned, we actually make an assumption that the elements of an example $(\mathbf{x}_t, y_t, \sigma_t)$ arrive at the same time. But in some practical applications, the label y_t is received after receiving training point $\mathbf{x}_t$ occasionally. There is no need to update the boundary vector after receiving all the elements of an example. Here, we propose a two-step update scheme.

The two-step update scheme has twice updates on each learning round. The first update takes place after the training point $\mathbf{x}_t$ arrives which updates the boundary vector using the geometry of the training points. The second update takes place after y_t, σ_t arrive which updates the boundary vector using the label. Obviously, EA update and overall update can be used in each update process of two-step update scheme. For example, we use EA update to describe the update process of two-step update scheme.

Denote as $\boldsymbol{\alpha}_{t-1/2}$ the coefficient vector after the first update on round t and $\boldsymbol{\omega}_{t-1/2}$ its associate boundary. The example-associate coefficients in the first update on round t are $\alpha_{t1}^1, \alpha_{t1}^2, \ldots, \alpha_{t(t-1)}^1, \alpha_{t(t-1)}^2$, and new associate boundary vector can be written as

$$\boldsymbol{\omega}_{t-1/2} = \boldsymbol{\omega}_{t-1} + 4\rho_{t-1/2} c_2^2 \sum_{i \in S_t} w_{ti}^2 \langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \rangle (\mathbf{x}_t - \mathbf{x}_i). \tag{34}$$

In the second update process, the example-associate coefficient is α_{t0}, and new associate boundary vector is

$$\boldsymbol{\omega}_t = \boldsymbol{\omega}_{t-1/2} + \rho_t c_1^2 \sigma_t y_t \left[1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1/2} \rangle\right]_+ \mathbf{x}_t. \tag{35}$$

If $\sigma_t = 0$, the second update process in (35) would not happen, and the two-step update degenerates into EA update. The range of $\rho_{t-1/2}$ and ρ_t can be obtained by the same process in Section 4.1. Similar as the previous analysis, the overall update also can be used in each update process of two-step update scheme.

The online MR algorithm based on the two-step update can be described in Algorithm 6.

This update scheme is more like a new perspective of online MR problem, and its effect is influenced by the update schemes on each step. Therefore, we pay more attentions to the first two update schemes aforementioned in this paper.

5. Sparse Approximations for Kernel Representation

In practice, kernel functions are always used to find a linear classifier, like SVM. Our online MR framework contains the product of two points, so we can easily introduce the kernel function in our framework. If we note K the kernel matrix such that

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

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

$$\boldsymbol{\omega}_t = \sum_{i=1}^{t} (\beta_i)_t \Phi(\mathbf{x}_i). \tag{37}$$

Unfortunately, the online MR algorithms with kernel functions in Section 4 have to store the example sequence up

INPUT: two positive scalars: c_1 and c_2; edge weights w_{ij}.
INITIALIZE: a coefficient vector $\boldsymbol{\alpha}_0$ and its associated decision boundary vector $\boldsymbol{\omega}_0$.
PROCESS: For $t = 1, 2, \ldots, T$
Receive an example $(\mathbf{x}_t, y_t, \sigma_t)$,
Get I_t using the process in Algorithm 4,
Choose a step size $\rho_t \in [0, \min\{\rho_t^{\max}, \rho_t^*\}]$,
Update the boundary vector using (33),
If $\sigma_t = 0$, predict $\widehat{y}_t = \text{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle)$,
Renew the buffer.

Algorithm 5: Online manifold regularization algorithm based on overall update.

INPUT: two positive scalars: c_1 and c_2; edge weights w_{ij}.
INITIALIZE: a coefficient vector $\boldsymbol{\alpha}_0$ and its associated decision boundary vector $\boldsymbol{\omega}_0$.
PROCESS: For $t = 1, 2, \ldots, T$
Receive a training point $\mathbf{x}_t$,
Choose a new coefficient vector $\boldsymbol{\alpha}_{t-(1/2)}$ that satisfies $D(\boldsymbol{\alpha}_{t-(1/2)}) \geq D(\boldsymbol{\alpha}_{t-1})$,
Return a new associated boundary vector $\boldsymbol{\omega}_{t-(1/2)}$ in (18).
Receive y_t, σ_t,
If $\sigma_t = 0$, $\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-(1/2)}$, $\boldsymbol{\omega}_t = \boldsymbol{\omega}_{t-(1/2)}$, predict $\widehat{y}_t = \text{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle)$;
Else if $\sigma_t = 1$, choose a new coefficient vector $\boldsymbol{\alpha}_t$ that satisfies $D(\boldsymbol{\alpha}_t) \geq D(\boldsymbol{\alpha}_{t-(1/2)})$, return a new associated boundary vector $\boldsymbol{\omega}_t$ in (18).

Algorithm 6: A template online manifold regularization algorithm based on two-step update.

to the current round (worst case). While using a buffering strategy for online MR which has a buffer size of τ, the stored matrix size is $\tau \times t$ and the time complexity is $O(\tau \times t)$ on round t. For practical purpose, we present two approaches to construct a sparse kernel representation for boundary vector on each round.

5.1. Absolute Threshold. To construct a sparse representation for the boundary vector, absolute threshold discards the examples whose associated coefficients are close to zero (more details in Section 7). Let $\varepsilon > 0$ denote the absolute threshold. When the absolute value of the associated coefficient of an input example $\mathbf{x}_i$ does not increase in further update process, $\mathbf{x}_i$ will be discarded if $|\beta_i| < \varepsilon$. The examples in the buffer cannot be discarded since the absolute values of their associated coefficients may increase in next rounds. The process of sparse approximation based on absolute threshold can be described in Algorithm 7.

The process of sparse approximation based on absolute threshold for different update schemes may be a little different in practical applications. For online MR algorithms with EA update, the coefficients of input examples which are not in the buffer will not change in further update process, and this sparse approximation process only deals with the example ($\mathbf{x}_{t-\tau}$ for Buffer-N) which is removed from the buffer on round t. For online MR algorithms with overall update, this sparse approximation process deals with all the examples which are not in the buffer on current round since the coefficients of these examples also can be changed. This approach may not work; if we are unlucky enough that all the $|\beta_i|$ are larger than ε on each round, the kernel representation of boundary vector will not become sparse at all.

5.2. k Maximal Coefficients (k-MC). Another way to construct a sparse kernel representation is to keep the examples of which the absolute value of associated coefficients are the first k maximum. This approach is called k maximal coefficients (k-MC) in this paper. Similar as the absolute threshold, k-MC does not discard the examples in the buffer of which absolute values of associated coefficients may increase in next round. The process of sparse approximation based on k-MC can be described in Algorithm 8.

While using k-MC for online MR algorithms which has a buffer size of τ, the stored kernel matrix size is at most $\tau \times (k + \tau)$ and the time complexity is $O(1)$ on each round.

6. On the Connection to Previous Work

6.1. About Dual Ascending Procedure. In the area of online learning, Shalev-Shwartz and Singer [13] propose a primal-dual perspective of online supervised learning algorithms. This work has the same dual ascending perspective as ours to achieve a better boundary vector. Different from it, we deal with an online MR problem of semisupervised learning, and our emphasis is how to construct a dual ascending model in semisupervised condition. An important conclusion in this paper is that the Fenchel conjugate of hinge functions is a key to transfer manifold regularization from offline to online, and this is also the reason why we use an absolute function to describe the difference between the predictions of two points.

INPUT: the absolute threshold ε; the kernel representation of boundary on round t: $\boldsymbol{\omega}_t = \sum_{i\in[t]} (\beta_i)_t \Phi(\mathbf{x}_i)$,
PROCESS: For each $\mathbf{x}_i$ in $\boldsymbol{\omega}_t$
If $\mathbf{x}_i$ is not in the buffer and $|(\beta_i)_t| < \varepsilon$, discard the example $\mathbf{x}_i$ and its associated coefficient $(\beta_i)_t$.
Return a new boundary $\boldsymbol{\omega}_t$.

Algorithm 7: The process of sparse approximation based on absolute threshold. This process only deals with the examples which will not be updated in the further update process.

INPUT: the parameter k; the kernel representation of boundary on round t: $\boldsymbol{\omega}_t = \sum_{i\in[t]} (\beta_i)_t \Phi(\mathbf{x}_i)$,
PROCESS: For each $\mathbf{x}_i$ in $\boldsymbol{\omega}_t$ and not in the buffer
If $(\beta_i)_t$ does not belong to the first k maximum of the coefficients, discard the example $\mathbf{x}_i$ and its associated coefficient $(\beta_i)_t$.
Return a new boundary $\boldsymbol{\omega}_t$.

Algorithm 8: The process of sparse approximation based on k-MC. The kernel representation for $\boldsymbol{\omega}_t$ contains $k + \tau$ examples at most in this condition, where τ is the buffer size.

The primal basic MR problem can degenerate into a basic supervised learning problem [14] while choosing the trade-off parameter $c_2 = 0$. Consider

$$J(\boldsymbol{\omega}) = \frac{1}{2}\boldsymbol{\omega}^2 + \sum_{t=1}^{T} c_1 [1 - y_t \langle \boldsymbol{\omega}, \mathbf{x}_t \rangle]_+. \tag{38}$$

Then, the dual function degenerates into

$$\begin{aligned} D(\boldsymbol{\alpha}) &= D(\alpha_{10}, \alpha_{20}, \ldots, \alpha_{T0}) \\ &= -\frac{1}{2}\left(\sum_{t=1}^{T} (c_1 \alpha_{t0} y_t \mathbf{x}_t)\right)^2 + \sum_{t=1}^{T} (c_1 \alpha_{t0}). \end{aligned} \tag{39}$$

Equation (39) is the dual function of basic supervised learning problem which is carefully discussed in [13].

6.2. About Online Manifold Regularization. Goldberg et al. [11] propose an empirical study of online MR which deals with the MR problem as follow:

$$\begin{aligned} J(\boldsymbol{\omega}) = \frac{1}{2}\boldsymbol{\omega}^2 &+ \frac{c_1}{l}\sum_{t=1}^{T} \sigma_t [1 - y_t \langle \boldsymbol{\omega}, \mathbf{x}_t \rangle]_+ \\ &+ \frac{c_2}{2T}\sum_{i,j=1}^{T} \left(\langle \boldsymbol{\omega}, \mathbf{x}_i \rangle - \langle \boldsymbol{\omega}, \mathbf{x}_j \rangle\right)^2 w_{ij}, \end{aligned} \tag{40}$$

where c_1 and c_2 are trade-off parameters and l is the number of labeled examples. Different from our framework, they use a square function to measure the difference between the predictions of two points (see Figure 2).

To avoid minimizing (40) directly, they further propose an *instantaneous regularized risk* $J_t(\boldsymbol{\omega})$ empirically on round t. Consider

$$\begin{aligned} J_t(\boldsymbol{\omega}) = \frac{1}{2}\boldsymbol{\omega}^2 &+ \frac{Tc_1}{l}\sigma_t [1 - y_t \langle \boldsymbol{\omega}, \mathbf{x}_t \rangle]_+ \\ &+ c_2 \sum_{i=1}^{t-1} \left(\langle \boldsymbol{\omega}, \mathbf{x}_t \rangle - \langle \boldsymbol{\omega}, \mathbf{x}_i \rangle\right)^2 w_{ti}. \end{aligned} \tag{41}$$

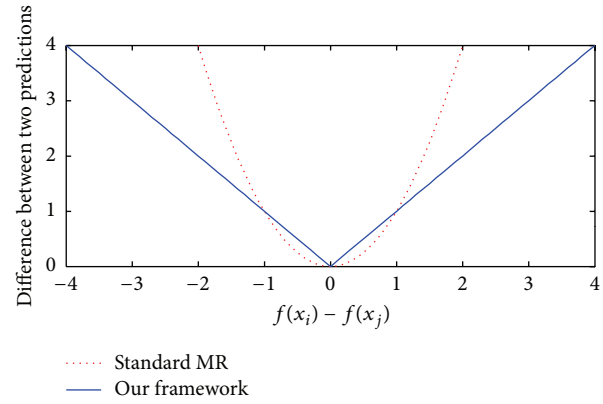

Figure 2: Different functions to measure the difference of prediction of two examples. Standard MR uses a square function, while our online MR framework uses an absolute function which can be decomposed into two hinge functions.

T/l is the reverse label probability $1/p_l$, which it assumes to be given and easily determined based on the rate at which humans can label the data at hand. In our work, we ignore this rate since it can be involved in the trade-off parameters c_1 and c_2.

Based on the notion that $\boldsymbol{\omega}_t$ has a form as $\boldsymbol{\omega}_t = \sum_{i=1}^{t} \beta_i \mathbf{x}_i$, Goldberg et al. perform a gradient descent step over $\boldsymbol{\omega}$ that aims to reduce the instantaneous risk $J_t(\boldsymbol{\omega})$ on each round. The update scheme can be written as

$$\begin{aligned} (\beta_t)_t &= \rho_t c_1 \sigma_t y_t \frac{T}{l}[1 - \langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \rangle]_+ \\ &\quad + 2\rho_t c_2 \sum_{i=1}^{t-1} w_{ti} \langle (\mathbf{x}_i - \mathbf{x}_t), \boldsymbol{\omega}_{t-1} \rangle, \\ (\beta_i)_t &= (1 - \rho_t)(\beta_i)_{t-1} - 2\rho_t c_2 w_{ti} \langle (\mathbf{x}_i - \mathbf{x}_t), \boldsymbol{\omega}_{t-1} \rangle, \\ &\quad i \in \{1, 2, \ldots, t-1\}. \end{aligned} \tag{42}$$

Furthermore, this work uses an annealing heuristic trick which chooses a decaying step size $\rho_t = \gamma/\sqrt{t}$, $\gamma = 0.1$. This online MR algorithm is an empirical result which demonstrates its practicability by experiments and does not have enough theoretical analysis.

Compared with previous work, our online MR framework reinterprets the online MR process based on the notion of ascending the dual function, and it also can be used to derive different online MR algorithms. Here, we demonstrate that the update scheme in (42) can be derived from our online MR framework.

In Section 4.2, the gradient direction $\mathbf{d}$ of overall update for ascending the dual function on round t can be written as

$$\mathbf{d} = \left[\frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial(\eta_t)_{t-1}}, \frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial(\alpha_{t0})_{t-1}}, \frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial(\alpha^1_{t1})_{t-1}}, \frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial(\alpha^2_{t1})_{t-1}}, \ldots, \frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial\left(\alpha^1_{t(t-1)}\right)_{t-1}}, \frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial\left(\alpha^2_{t(t-1)}\right)_{t-1}} \right]. \tag{43}$$

While choosing

$$\mathbf{d}' = \left[1, \frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial(\alpha_{t0})_{t-1}}, \frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial(\alpha^1_{t1})_{t-1}}, \frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial(\alpha^2_{t1})_{t-1}}, \ldots, \frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial\left(\alpha^1_{t(t-1)}\right)_{t-1}}, \frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial\left(\alpha^2_{t(t-1)}\right)_{t-1}} \right], \tag{44}$$

we have

$$\left\langle \mathbf{d}, \mathbf{d}' \right\rangle = \frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial(\eta_t)_{t-1}} + \left(\frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial(\alpha_{t0})_{t-1}} \right)^2 + \sum_{i=1}^{t-1} \left(\left(\frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial(\alpha^1_{ti})_{t-1}} \right)^2 + \left(\frac{\partial D(\boldsymbol{\alpha}_{t-1})}{\partial(\alpha^2_{ti})_{t-1}} \right)^2 \right). \tag{45}$$

If $\langle \boldsymbol{\omega}_{t-1}, \boldsymbol{\omega}_{t-1} \rangle - \sum_{i=1}^{t-1} c_1 (\alpha_{i0})_{t-1} \sigma_i > 0$, we must have $\langle \mathbf{d}, \mathbf{d}' \rangle > 0$ and $\mathbf{d}'$ is a feasible ascending direction to make $D(\boldsymbol{\alpha}_t) \geq D(\boldsymbol{\alpha}_{t-1})$. Using $\mathbf{d}'$ to ascend the dual function, the update scheme can be written as

$$\begin{aligned} (\beta_t)_t &= \rho_t c_1^2 \sigma_t y_t \left[1 - \left\langle y_t \mathbf{x}_t, \boldsymbol{\omega}_{t-1} \right\rangle \right]_+ \\ &\quad - 4\rho_t c_2^2 \sum_{i \in S_t} w_{ti}^2 \left\langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \right\rangle, \\ (\beta_i)_t &= (1-\rho_t)(\beta_i)_{t-1} + 4\rho_t c_2^2 w_{ti}^2 \left\langle (\mathbf{x}_t - \mathbf{x}_i), \boldsymbol{\omega}_{t-1} \right\rangle, \quad i \in S_t. \end{aligned} \tag{46}$$

Equations (42) and (46) are essentially the same update scheme with different trade-off parameters and edge weights.

7. Experiments and Analyses

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

Table 1: Different datasets in our experiments. These datasets have different properties which contain number of classes, dimensions, and size.

Dataset	Classes	Dims	Points
Two moons	2	2	4000
Two rotating spirals	2	2	8000
Isolet	2	617	3119
USPS	10	100	7191

7.1. Datasets and Protocols. We report experimental results on two artificial and two real-world datasets in Table 1 with different properties.

The artificial datasets consist of two-class problems. The generated method of two moons dataset is available at http://manifold.cs.uchicago.edu/manifold_regularization/manifold.html; we set the radius of two moons to 4 and the width to 2, and only one example for each class is labeled in this dataset. To demonstrate that our online MR can handle concept drift, we also perform our experiments on two rotating spirals dataset of which 2% examples are labeled. Figure 3 shows that the spirals smoothly rotate 360° during the sequence, and the target boundary drifts with the sequence of examples.

The real-world datasets consist of two-class and multiclass problems. The Isolet dataset derives from the Isolet database of letters of the English alphabet spoken in isolation (available from the UCI machine learning repository). The database contains utterances of 150 subjects who spoke the name of each letter of the English alphabet twice. The speakers are grouped into 5 sets of 30 speakers each, referred to as isolet1 through isolet5. We considered the task of classifying the first 13 letters of the English alphabet from the last 13 only using isolet1 and isolet5 (1 utterance is missing in isolet5 due to poor recording). During the online MR process, all 52 utterances of one speaker are labeled and all the rest are left unlabeled. Our USPS dataset contains the USPS training set on handwritten digit recognition (preprocessed using PCA to 100 dimensions), and we apply online MR algorithms to 45 binary classification problems that arise in pairwise classification; 5 examples are randomly labeled for each class.

Our experimental protocols are as the following.

(1) The training sequences are generated randomly from each datasets (except for two rotating spirals).

(2) The offline MR algorithm for comparison is a state-of-the-art semisupervised learning algorithm based on manifold regularization which is called LapSVM [12].

(3) Each example in each dataset is trained once during online MR process.

To avoid the influence of different training sequences, all results on each dataset are the average of five such trials except

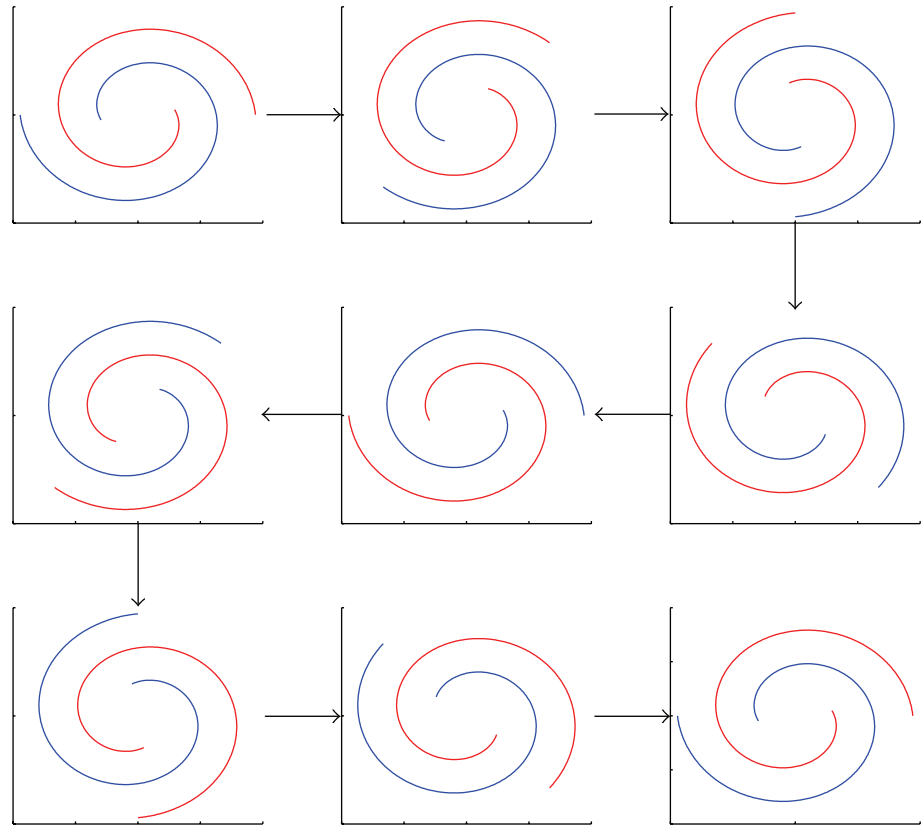

Figure 3: Two rotating spirals data sequence. We spin the two spirals dataset in top left during the sequence so that the spirals smoothly rotate 360° in every 8000 examples.

for two rotating spirals (this idea is inspired by [11]). The error bars are ±1 standard deviation.

All methods use the standard RBF kernel $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2(\sigma_K)^2}$. The edge weights are Gaussian weights which define a fully connected graph, and the edge weight parameter is σ. For online MR algorithms comparisons, we choose Buffer-N with $\tau = 200$ to avoid high computational complexity. We implemented all the experiments using MATLAB.

7.2. Computational Complexity. For offline MR, a $t \times t$ kernel matrix needs to be stored and inverted on round t, and the time complexity approximately amounts to $O(t^3)$ if using a gradient descent algorithm. Different from it, the computational complexity of our online MR algorithms is determined by the buffer size τ and the number of examples in the kernel representation of boundary vector on each round.

For our online MR without buffering strategies and sparse approximation approaches, the number of examples in the kernel representation is t, and the time complexity is $O(t^2)$. While using a buffering strategy for online MR which has a buffer size of τ, the time complexity reduces to $O(\tau \times t)$, but the number of examples in the kernel representation is still t. In practice, only part of the examples have to be stored (and computed) based on the sparse approximation. Figure 4 shows the number of examples in the kernel representation of boundary vector on each learning round for different sparse approximation approaches.

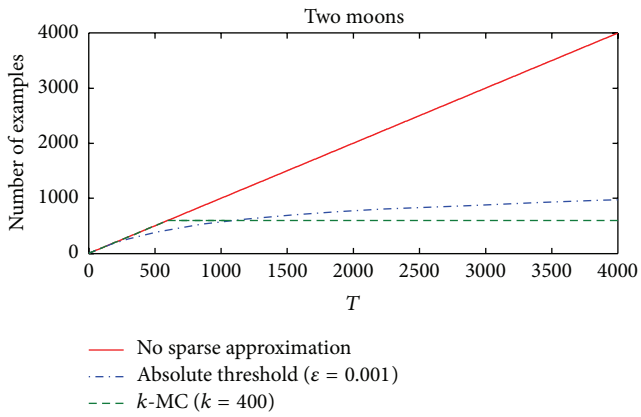

Figure 4: The number of examples in the kernel representation of boundary vector for different sparse approximation approaches. This experiment is on the two moons dataset which has 4000 examples. If no sparse representation approaches are used in the online MR, the kernel representation contains all the input examples. The number of examples in the kernel representation of boundary vector increases slowly while using an absolute threshold, and the number is at most $(k + \tau)$ while using k-MC for online MR algorithms.

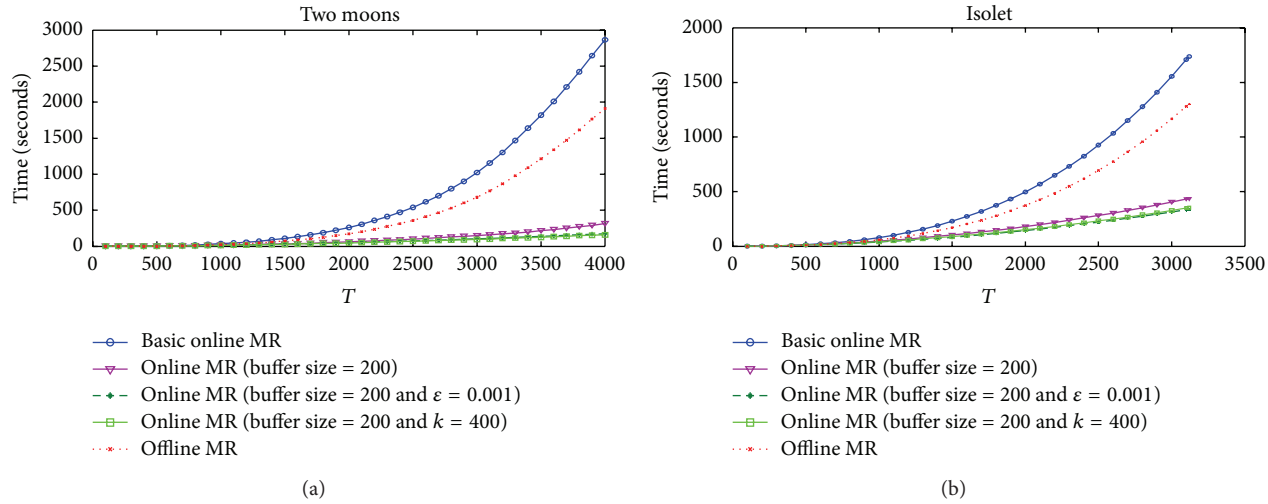

Figure 5: Cumulative runtime growth curves. (a) Experiments on two moons dataset, we generate a dataset which contains 4000 examples. (b) Experiments on Isolet dataset, this dataset has a high dimension. The curves have the similar trends on different datasets. Online MR algorithms with buffering strategies and sparse representation perform better than the others on the growth rate.

We also compare cumulative runtime curves of five different MR algorithms on the two moons and Isolet datasets. The first one is basic online MR which only uses ρ-EA update, but no buffering strategies and sparse approximation approaches. The second one is online MR which uses ρ-EA update and Buffer-N ($\tau = 200$). The third one is online MR which uses ρ-EA update, Buffer-N ($\tau = 200$), and an absolute threshold $\varepsilon = 0.001$. The fourth is online MR which uses ρ-EA update, Buffer-N ($\tau = 200$), and k-MC ($k = 400$). The last one uses offline MR (LapSVM) on each round. Figure 5 shows that online MR with buffering strategies and sparse representation performs better than basic online MR and offline MR on the runtime growth rate. Online MR algorithms without buffering strategies and sparse approximation approaches are time consuming and memory consuming, and it is intractable to apply them to real-world long time tasks.

The cumulative runtime growth curves of online MR with buffering strategies and sparse approximation approaches scale only linearly, while the others scale quadratically.

7.3. Accuracies. We used the same model selection strategy both for our online MR framework and traditional offline MR algorithms.

Based on the idea of "*interested in the best performance and simply select the parameter values minimizing the error*" [15], we select combinations of the parameter values on a finite grid in Table 2, and it is sufficient to perform algorithm comparisons.

While choosing an update scheme based on our online MR framework, we still have to select a step size ρ_t on each learning round. We report the online MR error rate for three scenarios in this paper.

Table 2: A finite grid of parameter values. We find the best performance of each online MR algorithm on this finite grid.

Parameter	Values
RBF width σ_K	$2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3$
Edge weight parameter σ	$2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3$
Penalty c_1	$10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3$
Penalty c_2	$10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3$

(i) Stationary step size $\rho_t = 0.1$.

(ii) Aggressive step size $\rho_t = \min\{\rho_t^{\max}, \rho_t^*\}$.

(iii) Decreasing step size $\rho_t = 0.1/\sqrt{t}$, which is also used in [11].

The best performances of all the online MR algorithms are presented in Table 3 and Figure 6. The following sections provide more additional details.

7.4. Additional Results. We now provide some additional results along the online MR algorithms run and discuss more precisely the effect of our derived online MR algorithms.

7.4.1. Effect of the Parameters σ_K, c_1, c_2 and the Step Size ρ. The parameters σ_K, c_1 and c_2 have similar effects on generalization as in the purely offline MR approach (see [12] for an empirical study). However, one has to try many choices of parameters during the model selection. The manifold regularizer incorporates unlabeled examples and causes the decision vector to appropriately adjust according to the geometry of training examples as c_2 is increased. If $c_2 = 0$, the unlabeled examples are disregarded and online MR degenerates into online supervised learning.

Table 3: Mean test error rates on different datasets. The error rates are reported for three different step size selection methods in the form of stationary step size/aggressive step size/decreasing step size. The result shows that our derived online MR algorithms achieve test accuracy comparable to offline MR. Specially, the experiments on two rotating spirals show that our online MR is able to track the changes in the sequence and maintain a much better error rate compared to offline MR. The performances of online MR algorithms are competitive with those of the state-of-the-art offline MR.

		Two moons	Two rotating spirals	Isolet
Offline MR	LapSVM	0	50	19.87
Online MR	EA update	2.60/6.75/11.92	2.45/13.31/46.45	20.13/31.15/26.60
	Overall update	3.43/8.40/10.80	0/0/19.50	20.38/31.22/24.36

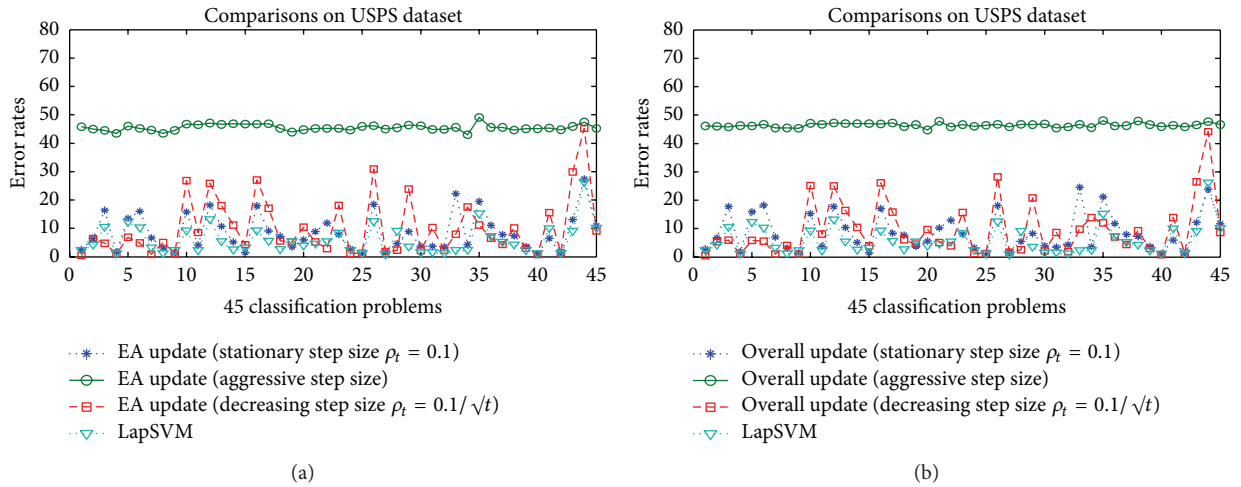

Figure 6: Mean test error rates for 45 binary classification problems on USPS dataset. The results show that the online MR with an aggressive step size does not perform well on this dataset, and the others achieve test accuracies that are comparable to LapSVM.

The step size ρ_t controls the increment of dual function $D(\boldsymbol{\alpha})$ on each learning round. We used three different step size selection methods for algorithm comparisons in last section. Here, we discuss the effect of different step size selection methods.

Stationary Step Size. Under mild conditions, this seemingly naive step size selection method has acceptable error rates on any input sequence. Figure 7 shows that a large stationary step size does not perform well in online MR algorithms. When one wishes to avoid optimizing the step size on each learning round, we suggest the stationary step size with a small value.

Aggressive Step Size. Since online MR algorithms adjust the boundary according to the local geometry of the incoming point and its label, the aggressive step size selection method aims to search for the optimal step size to increase the dual function more aggressively on each learning round. The experiments in Table 3 and Figure 6 imply that the aggressive selection method does not perform well on all the sequences.

Decreasing Step Size. This step size selection method is based on an idea that the boundary vector is approaching the optimal boundary as the online MR algorithms run. This selection method performs well on the datasets whose target boundaries are fixed, but the experiments on the two spirals dataset show that it does not perform well for drifting target boundaries.

7.4.2. Increasing Dual Function $D(\boldsymbol{\alpha})$ Achieves Comparable Risks and Error Rates. We compare the primal objective function $J(\boldsymbol{\omega}_t)$ versus the dual function $D(\boldsymbol{\alpha}_t)$ on the training sequence of two moons dataset as t increases. Figure 8 shows that the two curves approach each other along the online MR process using EA update ($\rho_t = 0.1$). The value of dual function $D(\boldsymbol{\alpha}_t)$ never decreases as t increases; correspondingly, the curve of primal function $J(\boldsymbol{\omega}_t)$ has a downward trend and some little fluctuations. Our experiments support the theory in Section 3 that increasing the dual problem achieves comparable risks of primal MR problem.

We also report the performance of $\boldsymbol{\omega}_t$ on the whole dataset in Figure 9. This result shows that the decision boundary is adjusted to be a better one along the online MR process. Since online MR adjusts the decision boundary according to the label of the incoming example and the local geometry of the buffer on each learning round, the error rate of $\boldsymbol{\omega}_t$ on the whole dataset is not always decreasing along the online MR process. It is also the reason why online MR can track the changes in the data sequence.

7.4.3. Online MR Handles Concept Drift. When the underlying distributions, both $P(\mathbf{x})$ and $P(y \mid \mathbf{x})$, change during

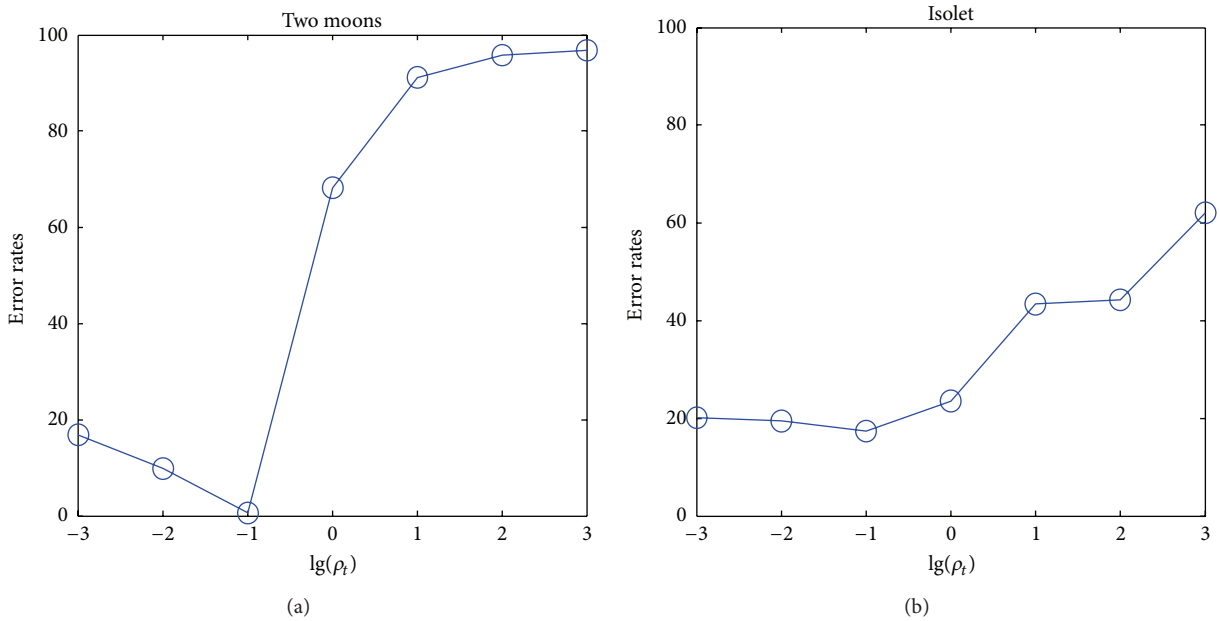

FIGURE 7: Error rates using different stationary step sizes. The parameters σ_K, c_1, c_2 in this experiment are all tuned for online MR algorithm using EA update and $\rho_t = 0.1$. The result implies that large step sizes lead to poor accuracies comparable to small ones.

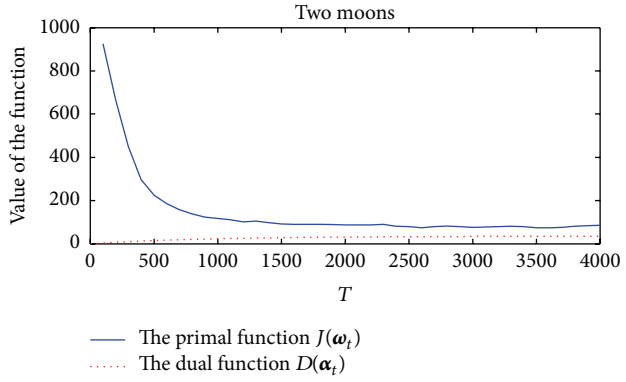

FIGURE 8: The value curves of the primal objective function $J(\boldsymbol{\omega}_t)$ and the dual function $D(\boldsymbol{\alpha}_t)$. The two curves approach each other as t increases.

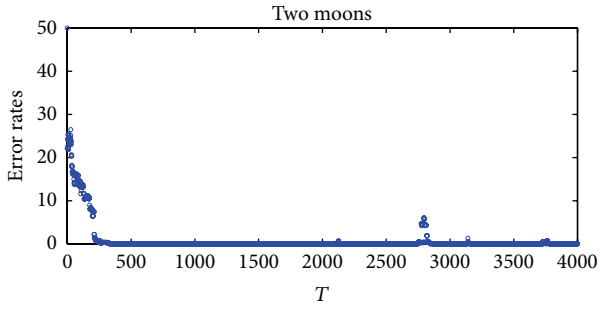

FIGURE 9: Error rates of $\boldsymbol{\omega}_t$ on the whole two moons dataset. The error rate of $\boldsymbol{\omega}_t$ has a downward trend, but it is not always decreasing along the online MR process.

the course of learning, the algorithms are expected to track the changes in the data sequence. In the two rotating spirals dataset, the points will change their true labels during the sequence and every stationary boundary vector will have an error rate of 50%.

We show the error rates of basic online MR versus online MR (Buffer-N) with different buffer sizes in Figure 10. This experiment illustrates that a suitable buffer size is able to adapt to the changing sequence and maintain a small error rate.

8. Conclusion and Future Directions

In this paper we presented an online manifold regularization framework based on dual ascending procedure. To ascend the dual function, we proposed three schemes to update the boundary on each learning rounds. Unfortunately, the basic online MR algorithms are time consuming and memory consuming. Therefore, we also applied buffering strategies and sparse approximation approaches to make online MR algorithms practical. Experiments show that our online MR algorithms can adjust the boundary vector with the input sequence and have risk and error rates comparable to offline MR. Specially, our online MR algorithms can handle the settings where the target boundary is not fixed but rather drifts with the sequence of examples.

There are many interesting questions remaining in the online semisupervised learning setting. For instance, we plan to study new online learning algorithms for other semisupervised regularizers those, in particular that with non-convex risks for unlabeled examples like S3VMs. Another direction is

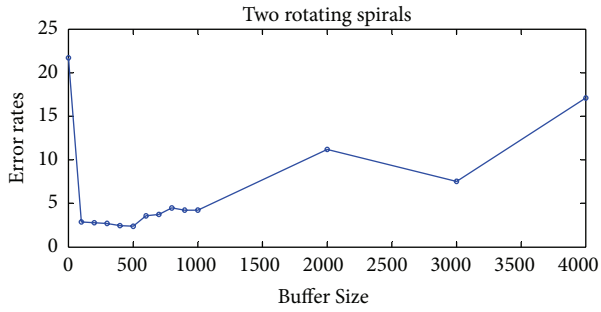

Figure 10: Error rates with different buffer sizes on two spirals data sequence. The buffer size can affect the capability of online MR to track the changes in the data sequence.

how to choose more effective parameters intelligently during the model selection.

Appendix

Fenchel Conjugate

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

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

Since f^* is defined as a supremum of linear functions, it is convex. The Fenchel-Young inequality states that for any $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$, we have $f(\boldsymbol{\omega}) + f^*(\boldsymbol{\lambda}) \geq \langle \boldsymbol{\lambda}, \boldsymbol{\omega}\rangle$.

Subgradients play an important role in the definition of Fenchel conjugate. In particular, the following lemma states that if $\boldsymbol{\lambda} \in \partial f(\boldsymbol{\omega})$, then Fenchel-Young inequality holds with equality. Here, we describe few lemmas of Fenchel conjugate which we use as theoretical tools in this paper. More details are in [16].

Lemma 2. *Let f be a closed and convex function, and let $\partial f(\boldsymbol{\omega})$ be its differential set at $\boldsymbol{\omega}$. Then, for all $\boldsymbol{\lambda} \in \partial f(\boldsymbol{\omega})$, one has $f(\boldsymbol{\omega}) + f^*(\boldsymbol{\lambda}) = \langle \boldsymbol{\lambda}, \boldsymbol{\omega}\rangle$.*

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

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

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

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

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

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

Combining the two inequalities, we have

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

□

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

Lemma 4. *Let $f(\boldsymbol{\omega}) = [b - \langle \boldsymbol{\omega}, \mathbf{x}\rangle]_+$ where $b \in \mathbb{R}_+$ and $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$ with $S = \mathbb{R}^{\mathbf{n}}$. Then, the conjugate of f is*

$$f^*(\boldsymbol{\lambda}) = \begin{cases} -\alpha b & \text{if } \boldsymbol{\lambda} \in \{-\alpha \mathbf{x} : \alpha \in [0,1]\} \\ \infty & \text{otherwise.} \end{cases} \quad (A.6)$$

Lemma 5. *Let f be a function, and let f^* be its Fenchel conjugate. For $a > 0$ and $b \in \mathbb{R}$, the Fenchel conjugate of $g(\boldsymbol{\omega}) = af(\boldsymbol{\omega}) + b$ is $g^*(\boldsymbol{\lambda}) = af^*(\boldsymbol{\lambda}/a) - b$.*

References

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

[2] K. Bennett and A. Demiriz, "Semi-supervised support vector machines," *Advances in Neural Information Processing Systems*, vol. 11, pp. 368–374, 1999.

[3] T. De Bie and N. Cristianini, *Semi-Supervised Learning Using Semi-Definite Programming*, MIT Press, Cambridge, Mass, USA, 2006.

[4] X. Zhu, "Semi-supervised learning literature survey," Tech. Rep. 1530, University of Wisconsin—Madison, 2008.

[5] B. Babenko, M. H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 983–990, Miami, Fla, USA, June 2009.

[6] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised online boosting for robust tracking," in *Proceedings of ECCV Computer Vision*, vol. 5302 of *Lecture Notes in Computer Science*, pp. 234–247, 2008.

[7] 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.

[8] D. Cohn, R. Caruana, and A. McCallum, "Semi-supervised clustering with user feedback," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2000.

[9] 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, 2002.

[10] 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.

[11] A. Goldberg, M. Li, and X. Zhu, "Online manifold regularization: a new learning setting and empirical study," in *Proceeding of ECML*, 2008.

[12] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: a geometric framework for learning from labeled and

unlabeled examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[13] S. Shalev-Shwartz and Y. Singer, "A primal-dual perspective of online learning algorithms," *Machine Learning*, vol. 69, no. 2-3, pp. 115–142, 2007.

[14] A. Blum, "On-line algorithms in machine learning," in *Proceedings of the Workshop on On-Line Algorithms*, A. Fiat and G. J. Woeginger, Eds., vol. 1442 of *Lecture Notes in Computer Science*, pp. 306–325, Springer, Berlin, Germany, 1998.

[15] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pp. 57–64, 2005.

[16] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, Berlin, Germany, 2nd edition, 1995.