Online Budgeted Least Squares with Unlabeled Data

Chen Huang*†, Peiyan Li*, Chongming Gao*, Qinli Yang* and Junming Shao*
*Data Ming Lab, University Of Electronic Science And Technology Of China, Chengdu, China
†JD Digits, Beijing, China
{huangc.uestc, peiyanli.uestc, chongming.gao}@gmail.com, {qinli.yang, junmshao}@uestc.edu.cn

***Abstract*—The scarcity of labeled data in real streaming environments has boosted the study of online semi-supervised learning (SSL). However, existing online SSL models often rely on some specific assumptions (e.g., manifold assumption) and need to maintain some extra constraints (e.g., the Laplacian matrix) on the fly, which is usually time and resource consuming. In this paper, we propose an efficient and effective online semi-supervised learning approach via Budgeted Least Square (BLS). Specifically, we first derive both closed-form transductive and inductive solutions for kernel least squares classification in the semi-supervised setting. Then, together with online kernel learning, BLS allows a concise online update. Besides, the theoretical regret bound of BLS is analysed, and empirical experiments on both static and streaming data further demonstrate its superiority over state-of-the-art algorithms.**

***Keywords*-semi-supervised learning; online Learning; classification**

I. Introduction

Although supervised learning has been successfully applied to many real-world applications, it requires to manually label all training data [1]. The problem becomes more challenging in the context of data streams. As a result, semi-supervised learning (SSL) has been introduced. By imposing some assumptions on unlabeled data (e.g., cluster assumption or manifold assumption), a large body of approaches have been proposed during the past decades. However, most SSL models often work on static data, being difficult to handle large-scale data sets. To design SSL models for large-scale streaming data is thus of significant importance.

Different from model re-training, online SSL targets to utilize both labeled and unlabeled instances to update the model on the fly. Considering the uncertainty of unlabeled data, many algorithms are equipped with online active learning or selective sampling [2]. In the interest of utilizing unlabeled data without resorting to their ground truth, one simple way is to regard the predicted/pseudo labels as their true labels [3]. However, such strategy could produce the cumulative error if unlabeled instances are incorrectly classified. To solve this problem, a mainstream idea is to utilize the structural information of unlabeled data, which can be broadly classified into two categories: cluster-based model and online manifold model. The cluster-based model tries to split arriving streaming data into several equal-sized chunks and then perform clustering algorithm on each chunk. Relying on the resulting clusters, the label of a continously arriving test instance is predicted based on some voting strategies among its neighboring clusters [4]. Such models are quite intuitive, but they strongly rely on the validity of the cluster assumption. Namely, the calculation cost and classification accuracy heavily depend on the chunk size and label purity of each cluster. As for the online manifold model, it is usually a kernel-based method with the adjacent information carried in an extra graph. Therefore, it often involves maintaining an adaptive graph (e.g., Laplacian matrix) during the whole online learning procedure. In addition, to ease the optimization, the Representer Theorem [5] is applied. It reduces the optimization problem in manifold regularization, into a finite dimensional problem of estimating the N expansion coefficients α_i [6]. Afterwards, to avoid that α grows linearly with data size N (Namely, the *curse of kernelization* [7]), budgeted kernel learning is applied [8]. However, the space complexity of online manifold model is usually quite large, because it requires to maintain a budgeted kernel matrix $\mathbf{K}$ and a Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N\times N}$(or a graph structure matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$) at the same time. This problem still exists, even if some approximation strategies are further fapplied [9]. Therefore, how to leverage the unlabeled instances to enhance classification performance effectively and efficiently in the online setting is still a big challenge.

In this paper, towards online SSL, we propose a novel online semi-supervised classification algorithm via Budgeted Least Square (BLS). Specifically, BLS is able to wrap unlabeled information into a data-dependent kernel on the fly, without constructing Laplacian matrix. Building upon semi-supervised least squares classification, we derive both closed-form transductive and inductive solutions. The former solution performs label propagation on the kernel matrix (instead of Laplacian matrix), while the latter, together with budgeted kernel learning [10], provides a concise way to update the model on the fly. Considering the label sparsity of data stream, we employ two budgets for labeled and unlabeled data. Also, two budget maintenance strategies are proposed to break the curse of kernelization, including (1) removing the oldest instance; (2) bounding the budget by projection. The basic idea of BLS is that once any budget overflows, the maintenance strategy would be triggered. Besides, an incremental update method is adopted to make

2374-8486/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDM.2019.00041

the calculation of model coefficients more tractable and the theoretical regret bound is also provided. Finally, extensive experimental results demonstrate that BLS not only outperforms the classical label propagation on the static datasets, but also gains a higher prediction performance in the online setting. The main contributions of this paper are as follows.

- **Novel Semi-supervised Least Square:** We first derive both the transductive and inductive closed-form solution of semi-supervised least squares, and then extend it into online setting. Also, we show our inductive solution is related to the harmonic solution of Label Propagation.
- **Budgeted Online Learning:** Resorting to budgeted kernel learning, BLS is able to fully capture the structural information of unlabeled data without extra constraints. Beyond, an incremental method is introduced to update the model, which makes BLS more scalable. Finally, the regret bound of BLS is also provided.

II. Related Work

Cluster-based Model. Building upon cluster assumption (i.e., instances belonging to the same cluster share the same label), cluster-based model performs clustering on both labeled and unlabeled data in each data chunk. To ensure the label purity of clusters, [11] employs semi-supervised clustering on each data chunk, while [4] targets to measure the cluster purity by information theory. Moreover, For the purpose of better modelling concept drift, incremental decision tree is applied in [12]. They employ simple K-means / K-mode to create clusters at leaves and those clusters are used as KNN classifiers and concept detectors. Finally, as soon as any test instance arrives, cluster-based model makes prediction by weighted voting among neighbourhood clusters. They update cluster weight by its classification performance on labeled data. However, the calculation cost and accuracy of KNN depend on the chunk size and label purity of each cluster. In addition, pseudo-labels of unlabeled instances are reused to update KNN model [13], which would bring about serious cumulative error, if unlabeled instances are predicted incorrectly.

Online Manifold Model. Online manifold model utilizes graph Laplacian $\mathbf{L} \in \mathbb{R}^{N\times N}$ to capture the structure information of unlabeled data, where N is the number of processed instances. On one hand, coarse graining [14] and data quantization [15] are applied to maintain a compact representation of the complete data adjacency graph. On the other hand, the Representer Theorem [5] $f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x})$ is applied to ease the online manifold optimization into a finite dimensional problem of estimating the N expansion coefficients α_i. For example, [6] optimizes online update solution in primal space, but it totally ignores the increasing size of Laplacian regularization and the kernelization. [16] solves online problem by dual ascending procedure and use buffering strategy to handle the curse of kernelization problem. But it requires the prepared data similarity graph $\mathbf{W} \in \mathbb{R}^{N\times N}$, which results in a huge calculation cost. To reduce the cost, [9] gives a SGD-based solution for semi-supervised support vector machine, where the gradient is approximated by uniformly edge sampling on graph $\mathbf{W}$. In summary, they need to maintain kernel expansion, but also cost extra effort to maintain data graph.

III. Proposed Method

Notations: We use lower case letters as scalars (e.g., α), lower case boldface letters as vectors (e.g., $\mathbf{f}$), upper case letters as elements of a matrix (e.g., U_{ij}) and boldface upper letters as matrices (e.g., $\mathbf{U}$). At i-th round, $\mathbf{x}_i \in \mathbb{R}^d$ describes an d-dimensional input feature vector with $y_i \in \{-1, +1\}$ being its label, if available. also, l and u are the number of labeled and unlabeled instances, respectively. Finally, notation $[\mathbf{A}, \mathbf{B}; \mathbf{C}, \mathbf{D}]$ means block matrix $\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$.

A. Semi-supervised Least Square and Novel Solutions

As a direct application of well-known least squares regression to the classification task, least squares classification fits model $\mathbf{f}(\mathbf{x}_i) = \mathbf{x}_i^T \mathbf{w}$ to encoded y_i (± 1) using standard least squares data fitting. It predicts the label of $\mathbf{x}_i$ by $sign(\mathbf{f}(\mathbf{x}_i))$. Formally, it optimizes the following problem.

$$\min_{\mathbf{f}} \|\widehat{\mathbf{y}} - \mathbf{f}\|_2^2 + \lambda \Omega(\mathbf{f}) \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{N\times d}$ and $\widehat{\mathbf{y}} \in \mathbb{R}^N$ are the input training data and corresponding labels, respectively. Here, $\mathbf{f} = \mathbf{X}\mathbf{w}$, where $\mathbf{w} \in \mathbb{R}^d$ is the model coefficient to learn and $\Omega(\mathbf{f})$ is the regularization term. In the semi-supervised setting, where data are partial labeled, semi-supervised least squares problem optimizes the following non-convex problem.

$$\min_{\mathbf{f},\mathbf{z}} \|\mathbb{Y} - \mathbf{f}\|_2^2 + \lambda \Omega(\mathbf{f}) \tag{2}$$

where $\mathbb{Y} = [\mathbf{y}; \mathbf{z}]$ and $\mathbf{z}$ is the predicted labels of unlabeled instances. By introducing the manifold assumption on data, Laplacian regularized least squares [17] is proposed by setting $\Omega(\mathbf{f}) = \mathbf{f}^T \mathbf{L} \mathbf{f} + ||\mathbf{f}||_2^2$. However, due to the space complexity of $\mathbf{L}$ grows quadratically as a function of number of instances, it is infeasible for large-scale data. To this end, by setting $\Omega(\mathbf{f}) = ||\mathbf{f}||_2^2$, semi-supervised least squares with no explicit assumption is recently getting more attention [18], [19]. However, those algorithms are limited to linear classification in the static environment. In this paper, towards online learning, we first enrich the expressiveness of semi-supervised least squares with kernel extension, and then we solve the problem by deriving a closed-form solution without iteration, which lends BLS to handling large-scale data sets in online setting.

In this paper, we assume that $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS) with a positive definite kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ implementing the inner product

$\langle \cdot, \cdot \rangle$. The inner product is defined so that it satisfies the reproducing property, $\langle k(\mathbf{x}, \cdot), f(\cdot) \rangle = f(x)$. Practically, the hypothesis $f(x)$ can be written as a kernel expansion over N training instances [5]. Namely, $f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x})$. Here, $N = l + u$. By adding a uniform prior on $\mathbf{z}$ to avoid the trivial solution, we rewrite Eq. (2) as follows. Here $\mathbf{K}$ is a kernel matrix.

$$\min_{\alpha, \mathbf{z}} \|\mathbb{Y} - \mathbf{K}\alpha\|_2^2 + \lambda_1 \alpha^T \mathbf{K} \alpha + \lambda_2 \|\mathbf{z}\|_2^2 \tag{3}$$

Note that Eq. (3) is equivalent to Eq. (2) when the linear kernel is chosen. To solve the problem, established methods such as [18], [19] optimize the model coefficient $\mathbf{w}$ and label $\mathbf{z}$ of Eq. (2) in an iterative fashion (i.e., fixing $\mathbf{z}$ (or $\mathbf{w}$), solve $\mathbf{w}$ (or $\mathbf{z}$) iteratively). Now, we show that a closed-form solution could be obtained by one simple trick. Given $\mathbf{z}$, the first order optimality condition for convex optimization yields:

$$\alpha = (\lambda_1 \mathbf{I} + \mathbf{K})^{-1} \mathbb{Y} \tag{4}$$

Inspired by [20], we formulate a quadratic approach, equivalent to Eq. (3). Basically, instead of regarding α as a constant when iteration optimization, we set $\alpha(\lambda_1) = (\lambda_1 \mathbf{I} + \mathbf{K})^{-1} \mathbb{Y}$ to be a variable, parameterized by λ_1. Note that $\alpha(\lambda_1)$ fully characterizes the information of variable $\mathbf{z}$. Now, plugging $\alpha(\lambda_1)$ back into to Eq. (3), we obtain a penalized quadratic problem only involving $\mathbf{z}$.

$$\min_{\mathbf{z}} \mathbb{Y}^T \widehat{\mathbf{K}} \mathbb{Y} + \lambda_2 \|\mathbf{z}\|_2^2 \tag{5}$$

where $\widehat{\mathbf{K}} = \mathbf{I} - \mathbf{K}(\lambda_1 \mathbf{I} + \mathbf{K})^{-1} = (\mathbf{I} + \mathbf{K}/\lambda_1)^{-1}$. Rewriting $\widehat{\mathbf{K}} = \begin{bmatrix} \widehat{\mathbf{K}}_{ll} & \widehat{\mathbf{K}}_{lu} \\ \widehat{\mathbf{K}}_{ul} & \widehat{\mathbf{K}}_{uu} \end{bmatrix}$, we derive the closed-form transductive and inductive solutions of semi-supervised least squares problem without any iteration.

$$\mathbf{z} = -(\widehat{\mathbf{K}}_{uu} + \lambda_2 \mathbf{I})^{-1} \widehat{\mathbf{K}}_{ul} \mathbf{y} \tag{6}$$

$$\alpha = \frac{1}{\lambda_1} \widehat{\mathbf{K}} [\mathbf{I}; -(\widehat{\mathbf{K}}_{uu} + \lambda_2 \mathbf{I})^{-1} \widehat{\mathbf{K}}_{ul}] \mathbf{y} \tag{7}$$

Note that Eq.(6) presents a transductive solution for semi-supervised least square problem. We remark that $\mathbf{z}$ does not involve α, it only relates to the training data. Interestingly, this solution is related to the standard label propagation approach to semi-supervised learning on graphs (See Section III-D1). On the other hand, Eq.(7) gives us an inductive solution, which is irrelevant to the pseudo labels $\mathbf{z}$ of unlabeled instances. It is optimized by fully exploiting the structural information of both labeled and unlabeled instances, which, in our case, refers to a better kernel construction. We remark that α is solved without introducing extra graph Laplacian constraint on unlabeled data.

Finally, kernelized semi-supervised least squares classifier is given by $sign(\mathbf{f}(\mathbf{x})) = sign(\alpha^T \mathbf{K}(\mathbf{X}, \mathbf{x}))$. Note that our proposed method is different from Manifold Regularization [21], which only derives the inductive solution for Laplacian regularized least squares. BLS gives both transductive and inductive solution for semi-supervised least squares without introducing the Laplacian matrix. Also, BLS is different from [22], which defines a data-dependent kernel on RKHS to penalize the complexity of semi-supervised model as a regularization term. In this paper we focus more on exploiting the modified kernel to explicitly learn the kernel coefficient α and predicted labels $\mathbf{z}$.

B. Budgeted Semi-supervised Least Squares

In this section, we extend BLS into the online setting and the budgeted semi-supervised least square algorithm is proposed to scale up to streaming data. Basically, after receiving one unseen example $\mathbf{x}_i$ at time i, BSLS makes prediction based on current model $\mathbf{f}_{i-1}$. Afterwords, BSLS refines $\mathbf{f}_{i-1}$ by updating the kernel coefficients α only, while online manifold model simultaneously updates model coefficients and calculates similarities between $\mathbf{x}_i$ and all (or part of) historical data. Given the curse of kernelization [7], we turn to budgeted kernel learning [10] to bound the growth of model size. In general, budgeted kernel learning is supervised. It learns the kernel-based predictive model as follows.

$$f(\mathbf{x}) = \sum_{i=1}^{B} \alpha_i k(\mathbf{x}_i, \mathbf{x})$$

Here, B is the size of budget. The existing budget online kernel classification approach aims to bound the number of support vectors by a budget constant B, using different budget maintenance strategies. In this paper, we maintain two budgets for semi-supervised online learning. Namely, B_L and B_U for labeled and unlabeled instances, respectively, where $B = B_L + B_U$.

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{B} \alpha_i k(\mathbf{x}_i, \mathbf{x}) \\ &= \sum_{i=1}^{B_L} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + \sum_{j=1}^{B_U} \alpha_j k(\mathbf{x}_j, \mathbf{x}) \end{aligned}$$

Accordingly, if the number of labeled (or unlabeled) instances in the budget overflows the pre-defined size, B_L (or B_U), a budget maintenance procedure is triggered to maintain the model size. Otherwise, we simply add the incoming instance into corresponding budget. However, this update strategy is too aggressive and results in too many updates once the budget is full. Therefore, we update the model only when prediction is wrong or the prediction confidence is low ($y_i f(\mathbf{x}_i) \leq 0$ or $|f(\mathbf{x}_i)| \leq \varepsilon$, where $\varepsilon \geq 0$ is a user-defined confidence threshold). To bound the model size, we present two budget maintenance strategies, including (1) removing the oldest sample; (2) merging $\mathbf{x}_i$ by projection.

Remove the Oldest. In this strategy, once the size of labeled or unlabeled part overflows its budget, the oldest instance in the corresponding budget would be simply discarded. Although it may change the hypothesis and decrease its accuracy, it turns out to be an efficient way [23]. Also, this strategy may be robust to concept drift, as it focuses more on current data. For better understanding, we give its pseudo-code in Algorithm 1.

Merging by Projection. Considering the information of the removed vectors are completely vanished, the idea of projection-based strategy is to preserve the information of the removed instance. Instances are not discarded, instead they are projected onto the space spanned by the previous online hypothesis. Specifically, we denote current budget size as b and current model $f = \sum_{i=1}^{b} \alpha_i k(\mathbf{x}_i, \cdot)$. Let $f' = \sum_{i=1}^{b+1} \alpha'_i k(\mathbf{x}_i, \cdot)$ be the model after adding new instance $\mathbf{x}_{b+1}$ into corresponding budget, $f'' = P_b f'$ denotes the projection of f' onto the space spanned by previous b budgeted instances. Here the projection P_b is an idempotent ($P_b^2 = P_b$) and linear operator.

Following [24], we obtain that $f'' = \sum_i^b \alpha'_i k(\mathbf{x}_i, \cdot) + \alpha'_{b+1} P_b k(\mathbf{x}_{b+1}, \cdot)$. Here, $P_b k(\mathbf{x}_{b+1}, \cdot) = \sum_j^b d_j k(\mathbf{x}_j, \cdot)$ and $(d_1, d_2, ..., d_b)$ is a set of coefficients to learn. To bound the model size, we use the projected hypothesis f'' as our next hypothesis if the distance δ between f' and f'' is small.

$$\begin{aligned}\|\delta\|_2^2 &= \|f'' - f'\|_2^2 \\ &= \left\|\alpha'_{b+1}(P_b k(\mathbf{x}_{b+1}, \cdot) - k(\mathbf{x}_{b+1}, \cdot))\right\|_2^2\end{aligned}$$

By minimizing the projection distance δ, we obtain the optimal coefficients $\mathbf{d}^* = \mathbf{K}_b^{-1}\mathbf{k}_{b+1}$, where

$$\begin{aligned}(\mathbf{K}_b)_{ij} &= k(\mathbf{x}_i, \mathbf{x}_j), i, j \in b \\ (\mathbf{k}_{b+1})_i &= k(\mathbf{x}_i, \mathbf{x}_{b+1})\end{aligned}$$

After substituting $\mathbf{d}^*$ into $\|\delta\|_2^2$, we have

$$\|\delta\|_2^2 = \alpha_{b+1}^{'2}(k(\mathbf{x}_{b+1}, \mathbf{x}_{b+1}) - \mathbf{k}_{b+1}^T \mathbf{d}^*)$$

Thus the projection-based budget is updated by using the following equation.

$$f'' = \sum_i^b \alpha'_i k(\mathbf{x}_i, \cdot) + \alpha'_{b+1} \sum_j^b \mathbf{d}^* k(\mathbf{x}_j, \cdot)$$

In detail, the budget size of projection-based strategy is not fixed, but is bounded (See Theorem 1 in [24]). We utilize the model f'' if δ is smaller than a given threshold τ. Otherwise, $\mathbf{x}_{b+1}$ would be added into corresponding budget, depending on whether it is labeled or not. Considering the sparsity of labeled data, we maintain an initial small budget B_L for labeled data. If B_L is full or the received instance is unlabeled, projection strategy is triggered. In summary, we show pseudo-code of projection strategy in Algorithm 2. Although projection strategy suffers extra computational cost to calculate $\mathbf{d}^*$, it achieves a better accuracy due to the reduced information loss. Finally, given two budgets and two maintenance strategies, the pseudo-code of BLS is given in Algorithm 3.

Algorithm 1 Remove Oldest Strategy

if the incoming instance $\mathbf{x}$ is labeled **then**
  Set $\mathbf{BM} = \mathbf{B_L}$;
else
  $\mathbf{BM} = \mathbf{B_U}$
end if
if $\mathbf{BM}$ is full **then**
  Remove the oldest instance from budget $\mathbf{BM}$;
end if
Add new $\mathbf{x}$ into budget $\mathbf{BM}$;

Algorithm 2 Projection Strategy

Require:
Projection threshold: τ

if $\mathbf{x}$ is labeled & B_L is not full **then**
  Add $\mathbf{x}$ into B_L;
else
  Calculate projection distance $\|\delta\|_2^2$;
  if $\|\delta\|_2^2 \leq \tau$ **then**
    Update budget by f'';
  else
    Add $\mathbf{x}$ into corresponding budget;
  end if
end if

Algorithm 3 Budgeted Least Square

Require:
Prediction Confidence: ε
Regularizations: λ_1, λ_2
Budget size: B_L, B_U
Budget strategy: S

for $t = 1$; $t \leq T$; t++ **do**
  Receive $\mathbf{x}_t$ and make prediction $sign(f(\mathbf{x}_t))$;
  if (y_t is available & $y_t f(\mathbf{x}_t) \leq 0$) || ($y_t$ is not available & $|f(\mathbf{x}_t)| \leq \varepsilon$) **then**
    Update budget by the strategy S;
  end if
end for

C. Efficient Update

Despite the effectiveness of BLS, it involves calculating the inverse of the kernel matrix $\mathbf{K}$, when updating the kernel coefficients α (and computing $\mathbf{d}^*$ for projection strategy). In order to enable a greater scalability of our algorithm, we introduce an efficient method to calculate it incrementally.

Given $\mathbf{K}_b^{-1}$, calculate $\mathbf{K}_{b+1}^{-1}$. This method is first introduced by [25]. In this paper, without loss of generality, we

denote $\mathbf{K}_b^{-1}$ as the inverse of kernel matrix $\mathbf{K}+\lambda\mathbf{I} \in R^{b\times b}$, where $\lambda \geq 0$, $\mathbf{x}_{b+1}$ is the new coming instance. After the addition of $\mathbf{x}_{b+1}$ into budget, the following equation holds

$$\mathbf{K}_{b+1}^{-1} = \begin{bmatrix} \mathbf{K}_b^{-1} & & 0 \\ & & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{1}{r_b}\begin{bmatrix} \mathbf{a}_b \\ -1 \end{bmatrix}\begin{bmatrix} \mathbf{a}_b^T & -1 \end{bmatrix} \quad (8)$$

, where $\mathbf{0} \in R^{b\times 1}$, $r_b = k(\mathbf{x}_{b+1}, \mathbf{x}_{b+1}) + \lambda - \mathbf{k}_{b+1}^T\mathbf{a}_b$ and $\mathbf{a}_b = \mathbf{K}_b^{-1}\mathbf{k}_{b+1}$. Note that $\mathbf{a}_b$ equals to the optimal coefficients $\mathbf{d}^*$ in projection strategy. Thus, without extra calculation, $\mathbf{d}^*$ is employed to update $\mathbf{K}_{b+1}^{-1}$ in Eq. (8). Also, the matrix $\mathbf{K}$ can be safely inverted since it is always full-rank by incremental construction. Lastly, owning to the incremental evaluation of Eq. (8), the time complexity of calculating the kernel inverse is reduced to $O(b^2)$.

Given $\mathbf{K}_b^{-1}$, Calculate $\mathbf{K}_{b-1}^{-1}$. Given $\mathbf{K}_b = \begin{bmatrix} k_{11} & \mathbf{k}_{2:b}^T \\ \mathbf{k}_{2:b} & \mathbf{K}_{b-1} \end{bmatrix}$, $\mathbf{K}_b^{-1} = \begin{bmatrix} g_{11} & \mathbf{g}_{2:b}^T \\ \mathbf{g}_{2:b} & \mathbf{G}_{b-1} \end{bmatrix}$, $k_{11}, g_{11} \in \mathbb{R}$ and $\mathbf{k}_{2:b}, \mathbf{g}_{2:b} \in \mathbb{R}^{b-1}$, The equation holds

$$K_{b-1}^{-1} = \mathbf{G}_{b-1} - \frac{\mathbf{g}_{2:b}\mathbf{g}_{2:b}^T}{g_{11}} \quad (9)$$

The equation presents an elegant way to calculate the inverse matrix after removing its first row and column. Therefore, we always put kernel information of the oldest instance in the first row and column of $\mathbf{K}_b$. Namely, $\mathbf{K}_b$ is calculated based on $\mathbf{X}^b = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_b]^T$, where $\mathbf{x}_1$ in the corresponding budget is the oldest, $\mathbf{x}_2$ is the second oldest, and so on. Unfortunately, if new data $\mathbf{x}_{b+1}$ is unlabeled and the oldest data is labeled, the incremental strategy would fail (The oldest unlabeled data should be removed in this case). Therefore, we need to remove arbitrary row and column of $\mathbf{K}_b$ incrementally. Inspired by [26], to remove instance $\mathbf{x}_i, i \geq 2$, we first exchange i-th row and column with the first by matrix transformation. Denote $\mathbf{P}_i \in \mathbb{R}^{b\times b}$ and $\mathbf{Q}_i \in \mathbb{R}^{(b-1)\times(b-1)}$ as follows.

$$\mathbf{P}_i = \begin{bmatrix} 0 & \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{i-2} & \mathbf{0} & \mathbf{0} \\ 1 & \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{b-i+1} \end{bmatrix}, \mathbf{Q}_i = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{i-1} & \mathbf{0} \\ 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{b-i} \end{bmatrix}$$

Note that $\mathbf{P}_i^{-1} = \mathbf{P}_i$ and $\mathbf{Q}_i^{-1} = \mathbf{Q}_i^T$. Here, $\mathbf{I}_j \in \mathbb{R}^{j\times j}$ is a unit matrix and $\mathbf{0}$ is the all-zeroes matrix of adequate dimensions. To remove arbitrary row and column of $\mathbf{K}_b^{-1}$, three steps are involved.

1) Calculate $\mathbf{T}_b^{-1} = \mathbf{P}_i\mathbf{K}_b^{-1}\mathbf{P}_i$ to exchange i-th row and column with the first.
2) Remove the first row and column of $\mathbf{T}_b^{-1}$ by E.q. (9)
3) Finally, $\mathbf{K}_{b-1}^{-1} = \mathbf{Q}_i\mathbf{T}_{b-1}^{-1}\mathbf{Q}_i$

D. Theoretical Analysis and Discussion

In this section, we analyse the connections between the BLS and other algorithms, and give its regret bound [27], which measures the performance of an online algorithm relative to the performance of a competing prediction mechanism

1) Connection to Label Propagation: The label propagation minimizes $\ell^T\mathbf{L}\ell$ under the constraints $\ell_i = y_i$ for all labeled data $\mathbf{x}_i$, where $\mathbf{L}$ is the un-normalized Laplacian of the data similarity graph $\mathbf{W}$ and ℓ is the vector of predictions. In this case, label propagation yields the *harmonic solution* [28] for unlabeled data.

$$\ell_u = -(\mathbf{L}_{uu})^{-1}\mathbf{L}_{ul}\mathbf{y}$$

Or *regularized harmonic solution* [29], where $\lambda > 0$ is a parameter.

$$\ell_u = -(\mathbf{L}_{uu} + \lambda\mathbf{I})^{-1}\mathbf{L}_{ul}\mathbf{y}$$

Suppose the eigen decomposition of $\mathbf{L}$ is $\sum_{i=1}^{N}\sigma_i\mathbf{v_i}\mathbf{v_i}^T$. *If $0 < \sigma_i < 1$ holds for all i, the transductive solution $\mathbf{z}$ of BLS is equivalent to the regularized harmonic solution when $\mathbf{K} = \lambda_1(\mathbf{L}^{-1} - \mathbf{I})$ (Namely when $\widehat{\mathbf{K}} = \mathbf{L}$).* However, it does not state that there is a underlying manifold assumption in our proposed method.

2) Regret Bound Analysis: Now, we analyse the performance of the BLS by providing its regret bound. Note that this bound can be applied to both two budget strategies. To ease the proof, we first rewrite model $f(\mathbf{x})$ to the following linear classification task on the new feature space, derived from the kernel approximation.

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{B}\alpha_i k(\mathbf{x}_i, \mathbf{x}) \\ &= \sum_{i=1}^{B}\alpha_i\phi(\mathbf{x}_i)^T\phi(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) \end{aligned}$$

Without loss of generality, we assume $\|\phi(\mathbf{x})\| \leq 1$. We denote $f(w)$ as the objective function of semi-supervised least square. Formally let C be a closed convex set with radius U, and $\mathbf{w}_1, ..., \mathbf{w}_T$ be a sequence of vectors such that $\mathbf{w}_1 \in C$ and $\mathbf{w}_{t+1} \leftarrow \prod_C(\mathbf{w}_t - \eta_t\nabla_t - \triangle_t)$, where ∇_t is the gradient of $f(\mathbf{w})$ at $\mathbf{w}_t$, η_t is a step size. Also, we define $\triangle_t$ as a vector, presenting the weight degradation caused by budget maintenance at t-th round. Finally $\prod_C(w)$ is a projection operation to C.

Theorem 1: Let $\mathbf{w}^*$ be the optimal solution of Eq. (3) after the kernel approximation. Define the gradient error $E_t = \triangle_t/\eta_t$ and assume $\|E_t\| \leq 1$. Define the average gradient error as $\overline{E} = \frac{1}{T}\sum_{t=1}^{T}\|E_t\|$ and average step size $\overline{\eta} = \frac{1}{T}\sum_{t=1}^{T}\eta_t$. Then, the following regret bound holds.

$$\begin{aligned} R(T) &= \frac{1}{T}\sum_{t=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{w}^*) \\ &\leq \frac{((2\lambda_1+1)U-2)^2}{2}\overline{\eta} + 2U\left\|\overline{E}\right\| \end{aligned}$$

Proof: Let $D_t = \|\mathbf{w}_t - \mathbf{u}\|^2 - \|\mathbf{w}_{t+1} - \mathbf{u}\|^2$ be the relative progress toward $\mathbf{u}$ at t-th round. By the definition of $\mathbf{w}_{t+1}$, we have

$$\begin{aligned} D_t &= \|\mathbf{w}_t - \mathbf{u}\|^2 - \left\| \prod_C (\mathbf{w}_{t+1} - \triangle_t) - \mathbf{u} \right\|^2 \\ &\geq \|\mathbf{w}_t - \mathbf{u}\|^2 - \|\mathbf{w}_{t+1} - \triangle_t - \mathbf{u}\| \\ &= -\eta_t^2 \|\partial_t\|_2^2 + 2\eta_t \nabla_t^T (\mathbf{w}_t - \mathbf{u}) + 2\eta_t E_t^T (\mathbf{w}_t - \mathbf{u}) \end{aligned}$$

, where $\partial_t = \nabla_t + E_t$ and it's bounded by

$$\begin{aligned} \|\partial_t\| &\leq \|\nabla_t\| + \|E_t\| \\ &= \|(2\lambda_1 + k(\mathbf{x}_t, \mathbf{x}_t))\mathbf{w}_t - \phi(\mathbf{x}_t)\widehat{y}_t\| + \|E_t\| \\ &\leq (2\lambda_1 + 1)U + 2 \end{aligned}$$

In addition, by applying the property of strong convexity of F, it follows:

$$\nabla_t^T (\mathbf{w}_t - \mathbf{u}) \geq f_t(\mathbf{w}_t) - f_t(\mathbf{u}) + \frac{\lambda_1}{2} \|\mathbf{w}_t - \mathbf{u}\|^2$$

Moreover, for any vector $\mathbf{a}$ and $\mathbf{b}$, we have $\frac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} \geq -1$. The following equation holds.

$$E_t^T (\mathbf{w}_t - \mathbf{u}) \geq -\|E_t\| \|\mathbf{w}_t - \mathbf{u}\| \geq -2U \|E_t\|$$

Thus, by setting $Q_t = f_t(\mathbf{w}_t) - f_t(\mathbf{u})$, we have:

$$Q_t \leq \frac{D_t}{2\eta_t} - \frac{\lambda_1}{2} \|\mathbf{w}_t - \mathbf{u}\|^2 + \frac{\eta_t}{2}((2\lambda_1+1)U+2)^2 + 2\|E_t\| U$$

Note that according to the definition of D_t, we could replace $\mathbf{w}^*$ with $\mathbf{u}$. Therefore, summing over all T, we have the upper bound of the cumulative regret $TR(T)$. As for the first two terms of the upper bound, we have

$$\begin{aligned} &\frac{1}{2} \sum_t^T \frac{D_t}{\eta_t} - \lambda_1 \|\mathbf{w}_t - \mathbf{w}^*\|^2 \\ &= \frac{1}{2}(\frac{1}{\eta_1} - \lambda_1) \|\mathbf{w}_1 - \mathbf{w}^*\|^2 + \frac{1}{2} \sum_{t=2}^T (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \lambda_1) \|\mathbf{w}_t - \mathbf{w}^*\|^2 \\ &- \frac{1}{2\eta_T} \|\mathbf{w}_{T+1} - \mathbf{w}^*\|^2 \end{aligned}$$

Here, we choose $\lambda_1 = max(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}, \frac{1}{\eta_1})$ for all t. Then the above equation is bounded by:

$$\frac{1}{2} \sum_t^T \frac{D_t}{\eta_t} - \lambda_1 \|\mathbf{w}_t - \mathbf{w}^*\|^2 \leq -\frac{1}{2\eta_T} \|\mathbf{w}_{T+1} - \mathbf{w}^*\|^2 \leq 0$$

Finally, we concludes the proof by plugging it back to the definition of $R(T)$.

$$\begin{aligned} R(T) &= \frac{1}{T} \sum_{t=1}^T F_t(\mathbf{w}_t) - F_t(\mathbf{w}^*) \\ &\leq \frac{((2\lambda_1 + 1)U - 2)^2}{2} \overline{\eta} + 2U \left\| \overline{E} \right\| \end{aligned}$$

Note that, instead of employing gradient descent, $\mathbf{w}_{t+1}$ in BLS is given by directly minimizing. So we utilize $\eta_t \nabla_t \approx \mathbf{w}_t - \mathbf{w}_{t+1} = \nabla \mathbf{w}$ and estimate η_t by minimizing $||\nabla \mathbf{w} - \eta \nabla_t||_2^2$, where $||\nabla \mathbf{w}|| \leq 2U$. The solution yields $\eta_t = \frac{\nabla_t^T \nabla \mathbf{w}}{||\nabla_t||}$. We further use λ_1 to control the L2 norm of $\mathbf{w}$ such that $\overline{\eta}$ should not be too large. Therefore, BLS has linear regret.

IV. Experiment

In this section, we establish quantitative experiments to prove the performance of BLS on real-world datasets. All datasets are available at the LIBSVM homepage[1], the SSL Benchmarks homepage[2] and the UCI Machine Learning Repository[3].

A. Transductive Learning On Static Datasets

Selection of Comparison Methods. Since BLS is connected to the Label Propagation, we conduct the transductive experiment and compare our algorithm with GFHF [28] (we denote it as LP), and regularized label propagation [29] (we denote it as RLP). Basically, LP and RLP perform the label propagation on the Laplacian matrix, while BLS performs on the kernel matrix.

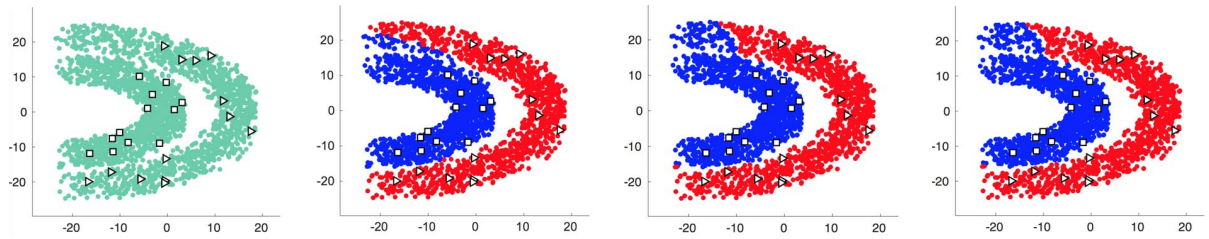
Experimental Setup. In our experiment, BLS uses linear kernel and RBF kernel with kernel width σ^2 being the average distance among instances. Also, we use the same RBF kernel to measure similarity matrix $\mathbf{W}$ for LP and RLP, which is a popular choose. As for the regularization parameters of RLP and BLSL, we tune in $\{10, 1, 0.1, 0.01\}$ and the best results are recorded. For all data sets, only 10% and 20% instances are randomly labeled and the rest remains unlabeled. Note that all comparing approaches work on the same data. Finally we run each method ten times to produce the average classification accuracy and standard deviation.

The experimental results are shown in Table I. Apparently, BLS outperforms both LP and RLP on most of datasets. One possible reason is that graph-based SSL, such as LP and RLP, capitalizes the abundance of unlabeled data based on the manifold assumption. However, this assumption may not hold on the real-world dataset and hence leads to a performance decrease. For a better illustration, we evaluate above methods on a synthetic data, shown in Fig 1. Here, two noisy manifold curves are generated and only 1% data are randomly labeled. Without parameter tuning, regularization parameters for all algorithms are set to be 1. We can observe that LP and RLP fail to obtain a good classification performance on the most noisy data region, while BLS yields a better result. Therefore, it is not surprised to see that BLS outperforms LP and RLP on both synthetic and real-word data sets.

B. Evaluation On Streaming Datasets

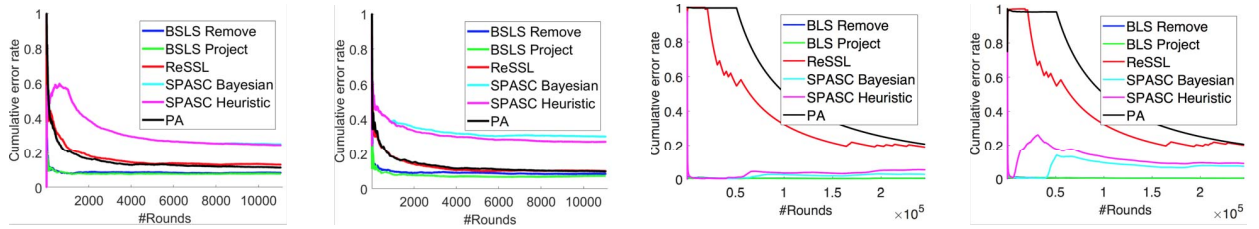
Selection of Comparison Methods. We compare with two recent online SSL approaches, **SPASC** [30] and **ReSSL**

[1]https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/
[2]http://olivier.chapelle.cc/ssl-book/benchmarks.html
[3]http://archive.ics.uci.edu/ml/



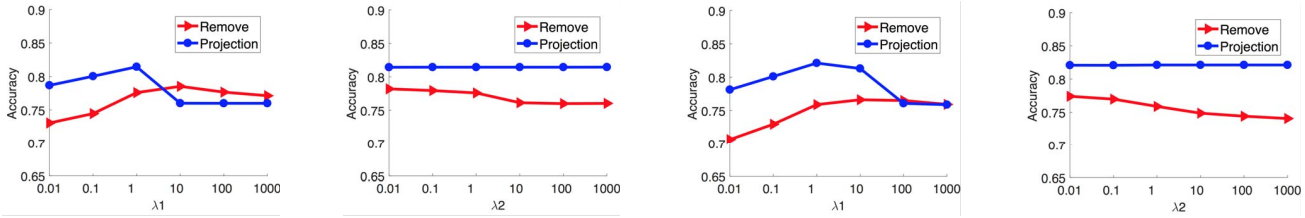
(a) Raw data (b) BLS (c) LP (d) RLP

Figure 1: Prediction illustration of different methods. Here the △ and the □ are labeled and the rest are unlabeled. Predicted classes are plotted in different color. Basically, LP and RLP fail to obtain a good classification performance on the most noisy data region, while BLS yields a better result.



(a) *phishing* + 10% labeled (b) *phishing* + 20% labeled (c) *skin* + 10% labeled (d) *skin* + 20% labeled data

Figure 2: Cumulative error rate of different kinds of model with 10% and 20% labeled data. Note that the lower the curve, the better the performance



(a) Tune λ_1 with 10% labels (b) Tune λ_2 with 10% labels (c) Tune λ_1 with 20% labels (d) Tune λ_2 with 20% labels

Figure 3: Sensitivity experiment on regularization parameters. We choose λ_1 (λ_2) from $\{0.01, 0.1, 1, 10, 100, 1000\}$, while λ_2 (λ_1) is fixed to be 1 and budget size is also fixed ($\mathbf{B_L} = 150, \mathbf{B_U} = 300$).

[4] and we also compare with one well-known supervised online model, **PA** [31], as a baseline.

Experimental Setup. To cope with the curse of kernelization problem, BLS needs to pre-define the budget size for $\mathbf{B_L}$ and $\mathbf{B_U}$. In our experiment, we simply set $\mathbf{B_L} = 150, \mathbf{B_U} = 300$. For projection strategy, we set threshold $\tau = log(t) * |f(\mathbf{x}_t)|$ such that τ reduces with more instances being observed. Also, if $|f(\mathbf{x}_t)|$ is low, $\mathbf{x}_t$ would have larger chance to be added into the budget. We set $\varepsilon = 0.25$ and use the same setting as last subsection for the rest parameters (Here, we randomly sampled 10000 instances to calculate their mean distance, used as σ^2).

As for others, PA is parameter-free. We tune batch size for SPASC in $\{50, 100, 500, 1000\}$ and the number of clusters in range from 5 to 50 with step size being 5. Furthermore, the parameters of ReSSL are set to be the value suggested in the paper. We repeat each experiment ten times to produce the average classification accuracy and standard deviation. Each time we randomly label 10% and 20% instances and all comparing approaches work on the same data.

The experimental results are presented in Table II. We mark the best result in bold font and underline the second best. We also show the results with respect to the round of online learning in Fig 2. Because of space limitations, we only show results on the *phishing* and *skin* datasets. In summary, compared with others, BLS gains a higher prediction performance. Interestingly, It seems that the projection-based strategy outperforms the remove-based strategy with smaller budget size, which indicates the superiority of retaining more information by projection strategy. Furthermore, we observe that the performance of BLS is always better than the baseline PA, while SPASC and ReSSL sometimes are worse. One possible reason, apart from the underlying difference between max margin model and KNN classifier,

Table I: The transductive accuracy (%) on 10% and 20% labeled data. Note that BLS is connected to the Label Propagation, therefore we compare our algorithm with GFHF [28] (denoted as LP), and regularized label propagation [29] (denoted as RLP). To be clear, we use bold font to mark the best results and underline the second best.

Data	Inst	Dim.	RBF	BLS		LP	RLP
				Linear			
Sonar	208	60	20%	**80.06±3.05**	71.99±2.81	58.61±9.49	68.55±4.59
			10%	**70.43±4.45**	67.11±3.62	50.59±6.61	59.79±7.76
Ionosphere	351	34	20%	**87.47±2.29**	82.31±1.96	63.81±0.75	63.81±0.75
			10%	**84.27±2.51**	81.20±1.49	63.92±0.87	64.15±1.11
Wdbc	569	30	20%	90.00±0.85	**94.62±0.93**	22.88±23.25	92.68±1.12
			10%	87.40±2.29	**92.93±1.24**	23.01±23.93	91.17±1.86
Australian	690	14	20%	**86.30±0.92**	86.29±1.20	66.85±13.68	77.84±10.23
			20%	85.01±1.28	**85.96±1.43**	60.85±11.38	74.40±12.72
Diabetes	768	8	20%	69.95±2.06	67.57±1.83	64.89±12.97	**70.99±1.59**
			10%	69.55±1.12	66.34±2.46	66.82±11.13	**70.84±1.82**
FourClass	862	2	20%	**89.10±1.49**	68.77±0.92	64.51±1.02	64.51±1.02
			10%	**88.08±2.92**	69.15±1.20	64.37±0.61	64.37±0.61
BankNote	1372	4	20%	**99.80±0.31**	94.62±0.73	90.86±3.60	97.76±0.84
			10%	**99.34±0.45**	94.40±0.83	76.06±11.55	94.46±3.86
G241n	1500	241	20%	78.64±5.31	**80.02±1.90**	56.85±9.31	78.14±5.72
			10%	76.56±3.07	**76.93±1.36**	55.48±9.92	76.35±3.28
G241c	1500	241	20%	76.01±7.31	**79.88±1.01**	55.62±9.12	75.57±7.89
			10%	70.47±6.50	**77.32±1.44**	52.30±7.41	69.89±6.86
Digit1	1500	241	20%	**97.25±0.48**	94.85±0.75	68.08±11.84	79.51±12.22
			10%	**95.40±0.95**	92.97±1.02	51.64±3.70	61.61±14.23
USPS	1500	241	20%	**95.43±0.53**	90.28±0.51	80.17±0.33	80.29±0.37
			10%	**92.87±1.19**	88.64±0.90	79.95±0.31	80.00±0.37
a1a	1605	123	20%	82.38±0.42	**82.51±0.52**	75.55±0.46	75.55±0.46
			10%	81.08±1.22	**81.37±0.91**	75.38±0.36	75.38±0.36
w1a	2477	300	20%	**97.52±0.27**	88.86±0.49	97.14±0.18	97.14±0.18
			10%	**97.36±0.29**	88.80±0.28	97.12±0.10	97.12±0.10
mushrooms	8124	112	20%	**99.97±0.05**	99.91±0.08	89.76±0.35	89.89±0.22
			10%	**99.77±0.25**	99.76±0.18	84.28±8.95	89.35±0.33

Table II: The online classification accuracy (%) on 20% and 10% labeled data. B_{proj} and B_{rm} indicate the total budget size (including the labeled and unlabeled budget, $\mathbf{B_L}$ and $\mathbf{B_U}$) of projection-based and remove-based strategy, respectively.

Data	Inst./Dim.		PA	ReSSL	SPASC		BLS (RBF)			
					Heuristic	Bayesian	Remove	B_{rm}	Project	B_{proj}
phishing	11055/68	20%	90.48±0.41	89.21±0.40	78.32±2.00	76.78±1.54	90.95±0.36	450	**92.49±0.25**	204.9±6.9
		10%	88.84±0.57	87.08±0.40	78.00±1.43	78.49±1.21	91.64±0.30	450	**92.08±0.34**	182.0±6.9
a6a	11220/123	20%	78.00±0.51	77.68±0.32	79.27±0.69	79.25±0.70	77.03±0.34	450	**81.31±0.73**	183.9±43.5
		10%	78.10±0.75	78.05±0.50	78.50±1.47	78.46±1.45	77.98±0.59	450	**80.83±0.97**	168.4±22.8
a7a	16100/123	20%	78.38±0.34	77.50±0.35	79.58±0.45	79.60±0.47	76.94±0.24	450	**81.22±0.76**	173.2±10.1
		10%	78.06±0.53	77.54±0.50	80.05±1.03	79.72±1.46	77.20±0.49	450	**80.83±0.79**	181.3±33.2
w6a	17188/300	20%	96.94±0.00	95.49±0.53	97.04±0.03	97.03±0.03	97.09±0.20	450	**97.13±0.15**	182.2±21.7
		10%	96.95±0.00	94.30±0.70	96.79±0.65	97.01±0.05	**97.21±0.09**	450	97.07±0.15	164.1±8.4
a8a	22696/123	20%	78.33±0.38	77.59±0.32	78.62±1.13	78.60±1.12	76.79±0.37	450	**81.59±0.67**	250.8±112.8
		10%	78.19±0.47	77.90±0.28	78.34±1.56	78.29±1.57	77.33±0.41	450	**81.44±0.74**	192.0±50.1
w7a	24692/300	20%	96.99±0.02	95.54±2.19	97.12±0.04	97.12±0.04	96.66±0.19	450	**97.28±0.12**	183.1±11.6
		10%	97.00±0.00	93.38±4.47	97.18±0.09	97.19±0.09	**97.24±0.12**	450	97.12±0.13	164.8±9.0
a9a	32561/123	20%	78.78±0.27	77.03±0.32	78.80±0.79	78.69±0.90	77.11±0.23	450	**82.09±0.33**	194.8±21.2
		10%	78.40±0.31	77.59±0.34	78.29±1.51	78.65±1.06	77.15±0.33	450	**81.51±0.61**	183.0±55.5
electricity	45312/8	20%	56.90±0.28	59.55±3.04	56.87±1.14	57.17±1.56	65.13±0.31	450	**68.39±3.46**	216.0±23.2
		10%	56.35±0.28	59.67±1.28	57.31±1.23	57.08±1.24	65.03±0.33	450	**68.97±1.73**	197.7±5.9
w8a	49749/300	20%	97.03±0.00	94.80±0.68	97.17±0.02	97.17±0.02	96.76±0.06	450	**97.28±0.16**	221.0±21.6
		10%	97.03±0.00	94.11±0.45	97.09±0.09	97.16±0.06	96.68±0.24	450	**97.24±0.15**	186.0±18.5
ijcnn1	49990/22	20%	89.38±0.14	92.08±0.31	89.86±0.05	89.89±0.04	92.66±0.24	450	**95.78±0.11**	529.0±14.9
		10%	91.64±0.30	91.23±0.16	89.15±0.56	89.26±0.59	92.76±0.31	450	**94.91±0.30**	375.9±12.9
cod-rna	59535/8	20%	76.80±0.64	**89.16±2.65**	72.91±2.28	75.71±1.52	79.19±0.23	450	85.64±0.68	185.0±61.1
		10%	75.69±0.51	**88.70±3.23**	75.32±1.88	75.60±2.14	79.46±0.42	450	86.17±0.57	153.0±2.4
skin	245057/3	20%	79.34±0.05	85.66±0.43	93.58±0.93	95.27±0.32	**99.73±0.01**	450	**99.73±0.01**	217.5±21.3
		10%	79.38±0.16	87.79±1.54	94.40±1.03	96.12±1.20	**99.62±0.01**	450	99.54±0.14	204.3±15.0

may be the inconsistency between real-word data and model assumption. In summary, BLS is more robust, and hence ensures the high empirical performance on real-word data sets.

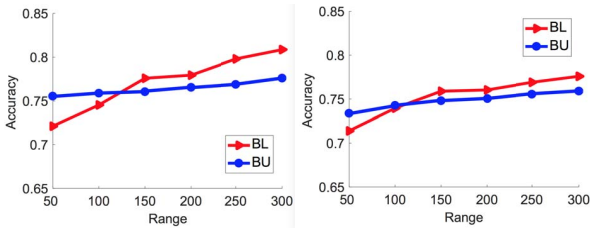
C. Sensitivity Analysis

Budget size. We investigate the influence of the budget size (i.e., $\mathbf{B_L}$ and $\mathbf{B_U}$) to the learning performance. Since the budget size of projection strategy is chosen adaptively, here we only vary the value of $\mathbf{B_L}$ and $\mathbf{B_U}$ for remove-based strategy. Basically, the bigger the budget, the more information we preserve. We show how the change of parameters influences the classification performance on the data set *a6a* with 10% (See Fig. 4(a)) and 20% labeled data (See Fig. 4(b)). Basically, we tune $\mathbf{B_L}$ (or $\mathbf{B_U}$) in $\{50, 100, 150, 200, 250, 300\}$ while fixing the value of $\mathbf{B_U} = 300$ (or $\mathbf{B_L} = 150$). Note that regularization parameters are set to be one in all experiments for simplicity. All experiments work on the same data and the prediction accuracy is recorded. It can be observed that when increasing the labeled budget size $\mathbf{B_L}$ or the unlabeled budget size $\mathbf{B_U}$, the classification accuracy tends to increase. The boost is more significant for $\mathbf{B_L}$, since more discriminative information are preserved to train a better classifier.

Regularization parameters. BLS has two regularization parameters, i.e. λ_1, punishing the model complexity and λ_2, which is a label prior parameter. We now investigate how performance change when λ_1 and λ_2 vary. Basically, we choose λ_1 (λ_2) to be $\{0.01, 0.1, 1, 10, 100, 1000\}$, while λ_2 (λ_1) is set to be 1 and budget size is also fixed ($\mathbf{B_L} = 150, \mathbf{B_U} = 300$). Figure 3 illustrates the performance of tuning λ_1 (See Fig. 3 (a) and (c)) and tuning λ_2 (See Fig. 3 (b) and (d)) on 10% and 20% labeled data. It can be observed that λ_1 is relatively more sensitive comparing to λ_2. With the growth of λ_1, we put larger penalty on model complexity. On one hand, too large value of λ_1 harms the expressiveness of our model, and hence hurt the accuracy. On the other hand, complex model tends to overfit budgeted instances. Theoretically, according to the proof of our regret bound, $\lambda_1 = max(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}, \frac{1}{\eta_1}), \forall t \in T$ may be a nice choose. but it requires to derive the mapping function for the user-specified kernel. Further more, λ_2 measures the label uncertainty and assigns uniform prior to unlabeled data. As is shown in the figure, projection based strategy is more robust to the change of λ_2. Generally, the absolute predicted value $|f(\mathbf{x})|$ inclines to be small if λ_2 is large. Since we use $\tau = log(t)|f(\mathbf{x})|$ to be the projection threshold, larger λ_2 could trigger more budget update, and hence more information is preserved to refine model on the fly. Thus, the selection of λ_1 remains to be a trade-off problem between overfitting and underfitting.

V. Conclusion

In this paper, towards online semi-supervised learning, we propose a novel budgeted semi-supervised least square for online classification, called BLS. Building on semi-supervised kernel least square, we derive its closed-form transductive and inductive solutions. The former is related to the regularized harmonic solution of the Label Propagation on kernel matrix, while the latter, together with budgeted kernel learning, provides a concise method to update the model without extra constraints on unlabeled data. Considering the label sparsity, we use two budgets for labeled and unlabeled data respectively and two budget maintenance strategies are proposed. Meanwhile, in order to more efficiently update the model on the fly, BLS adopts an incremental method to calculate the inverse of the kernel matrix. Also, we derive the theoretical regret bound of BLS for the comprehensive understanding. Finally, extensive experimental results demonstrate that BLS not only out-performs the classical label propagation on the static datasets, but also gains a higher prediction performance in the online experiments.



(a) On 10% labels (b) On 20% labels

Figure 4: Sensitivity experiments when varying the size of the two budgets. Since the budget size of projection strategy is chosen adaptively, we only vary the value of $\mathbf{B_L}$ and $\mathbf{B_U}$ for remove-based strategy.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (61976044, 41601025), Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (161062), National key research and development program (2016YFB0502300).

References

[1] J. Shao, F. Huang, Q. Yang, and G. Luo, "Robust prototype-based learning on data streams," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 5, pp. 978–991, 2017.

[2] A. B. Goldberg, X. Zhu, A. Furger, and J.-M. Xu, "Oasis: Online active semi-supervised learning." in *AAAI Conference on Artificial Intelligence*, 2011.

[3] Y. Haimovitch, K. Crammer, and S. Mannor, "More is better: Large scale partially-supervised sentiment classification," in *Asian Conference on Machine Learning*, 2012.

[4] J. Shao, C. Huang, Q. Yang, and G. Luo, "Reliable semi-supervised learning." in *IEEE International Conference on Data Mining*. IEEE, 2016, pp. 1197–1202.

[5] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *International conference on computational learning theory*, 2001, pp. 416–426.

[6] Y. Jia, S. Yan, and C. Zhang, "Semi-supervised classification on evolutionary data." in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009, pp. 1083–1088.

[7] Z. Wang, K. Crammer, and S. Vucetic, "Breaking the curse of kernelization: Budgeted stochastic gradient descent for large-scale svm training," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3103–3131, Oct. 2012.

[8] A. B. Goldberg, M. Li, and X. Zhu, "Online manifold regularization: A new learning setting and empirical study," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2008, pp. 393–407.

[9] T. Le, P. Duong, M. Dinh, T. D. Nguyen, V. Nguyen, and D. Q. Phung, "Budgeted semi-supervised support vector machine," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2016.

[10] J. Lu, S. C. Hoi, J. Wang, P. Zhao, and Z.-Y. Liu, "Large scale online kernel learning," *The Journal of Machine Learning Research*, vol. 17, no. 47, pp. 1–43, 2016.

[11] M. M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham, "A practical approach to classify evolving data streams: Training with limited amount of labeled data," in *2008 Eighth IEEE International Conference on Data Mining*, Dec 2008, pp. 929–934.

[12] P. Li, X. Wu, and X. Hu, "Mining recurring concept drifts with limited labeled streaming data," in *Proceedings of 2nd Asian Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 13. Tokyo, Japan: PMLR, 08–10 Nov 2010, pp. 241–252.

[13] A. Haque, L. Khan, and M. Baron, "Sand: Semi-supervised adaptive novel class detection and classification over data stream." in *The Association for the Advancement of Artificial Intelligence (AAAI)*, 2016, pp. 1652–1658.

[14] M. Farajtabar, A. Shaban, H. R. Rabiee, and M. H. Rohban, *Manifold Coarse Graining for Online Semi-supervised Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 391–406.

[15] B. Kveton, M. Philipose, M. Valko, and L. Huang, "Online semi-supervised perception: Real-time learning without explicit feedback," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, June 2010, pp. 15–21.

[16] B. Sun, G. Li, L. Jia, and H. Zhang, "Online manifold regularization by dual ascending procedure," *Mathematical Problems in Engineering*, 2013.

[17] J. Chen, C. Wang, Y. Sun, and X. S. Shen, "Semi-supervised laplacian regularized least squares algorithm for localization in wireless sensor networks," *Computer Networks*, 2011.

[18] D. Wang, F. Nie, and H. Huang, "Large-scale adaptive semi-supervised learning via unified inductive and transductive model," ser. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, 2014, pp. 482–491.

[19] J. H. Krijthe and M. Loog, "Robust semi-supervised least squares classification by implicit constraints," *Pattern Recognition*, vol. 63, 2017.

[20] S. Lv and H. Lian, "A debiased distributed estimation for sparse partially linear models in diverging dimensions," *arXiv preprint arXiv:1708.05487*, 2017.

[21] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, 2006.

[22] S. Vikas, N. Partha, and B. Mikhail, "Beyond the point cloud: from transductive to semi-supervised learning," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 824–831.

[23] O. Dekel, S.-s. Shai, and Y. Singer, "The forgetron: A kernel-based perceptron on a fixed budget," in *Advances in neural information processing systems*, 2006, pp. 259–266.

[24] F. Orabona, J. Keshet, and B. Caputo, "Bounded kernel-based online learning," *Journal of Machine Learning Research*, vol. 10, pp. 2643–2666, Dec. 2009.

[25] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in neural information processing systems*. MIT Press, 2001.

[26] S. V. Vaerenbergh, I. Santamara, W. Liu, and J. C. Prncipe, "Fixed-budget kernel recursive least-squares," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010.

[27] S. Shalev-Shwartz, "Online learning: Theory, algorithms, and applications," 08 2007.

[28] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," ser. Proceedings of the 20th International conference on Machine learning (ICML-03), 2003.

[29] B. Kveton, M. Valko, A. Rahimi, and L. Huang, "Semi-supervised learning with max-margin graph cuts," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

[30] M. J. Hosseini, A. Gholipour, and H. Beigy, "An ensemble of cluster-based classifiers for semi-supervised classification of non-stationary data streams," *Knowledge and information systems*, vol. 46, no. 3, pp. 567–597, 2016.

[31] S. Shalev-Shwartz, K. Crammer, O. Dekel, and Y. Singer, "Online passive-aggressive algorithms." in *Advances in neural information processing systems*, 2003, pp. 1229–1236.