Semisupervised Negative Correlation Learning

Huanhuan Chen, *Senior Member, IEEE*, Bingbing Jiang, and Xin Yao, *Fellow, IEEE*

***Abstract*— Negative correlation learning (NCL) is an ensemble learning algorithm that introduces a correlation penalty term to the cost function of each individual ensemble member. Each ensemble member minimizes its mean square error and its error correlation with the rest of the ensemble. This paper analyzes NCL and reveals that adopting a negative correlation term for unlabeled data is beneficial to improving the model performance in the semisupervised learning (SSL) setting. We then propose a novel SSL algorithm, Semisupervised NCL (SemiNCL) algorithm. The algorithm considers the negative correlation terms for both labeled and unlabeled data for the semisupervised problems. In order to reduce the computational and memory complexity, an accelerated SemiNCL is derived from the distributed least square algorithm. In addition, we have derived a bound for two parameters in SemiNCL based on an analysis of the Hessian matrix of the error function. The new algorithm is evaluated by extensive experiments with various ratios of labeled and unlabeled training data. Comparisons with other state-of-the-art supervised and semisupervised algorithms confirm that SemiNCL achieves the best overall performance.**

***Index Terms*— Committee machines, ensemble learning, multiple classifiers, negative correlation learning, semi-supervised learning.**

I. Introduction

SEVERAL machine learning paradigms have been developed for incorporating unlabeled examples into the supervised learning process, such as *semisupervised learning* (SSL) [52], *transductive learning* [31], etc. Both the SSL and transductive learning attempt to directly exploit unlabeled examples. While in SSL the unlabeled examples are typically different from the test examples, the unlabeled examples are exactly the same as the test examples in transductive learning.

Ensembles can improve the generalization performance when compared with a single learner in both supervised [34] and SSL [14], [48], [50]. In this paper, we propose a novel semisupervised algorithm, *semisupervised negative correlation learning* (SemiNCL), by generating a group of *diverse* learners from the *same* input space. This algorithm is based on NCL [34], [35], which is a specific ensemble training algorithm by managing the tradeoff between accuracy and diversity among base learners [36], [47].

It is well acknowledged that the generalization of an ensemble is related to the accuracy (measured by the training error) of the base learners and the diversity among them [9], [46]. Generally, a higher average accuracy of base learners and a larger diversity among base learners can lead to a better ensemble [9], [48]. However, it is a dilemma to simultaneously optimize both the accuracy and diversity in an ensemble. For example, a higher accuracy of base learners means most of ensemble members perform correctly on labeled data, therefore, a higher diversity is difficult to achieve. Fortunately, unlabeled data can be employed to promote diversity without degrading accuracy on the labeled data. Therefore, a better generalization could be achieved [50].

SemiNCL is based on this idea by promoting diversity utilizing both the labeled and unlabeled data. SemiNCL introduces a correlation penalty term on both labeled and unlabeled data into the cost function of each ensemble member, so that each ensemble member minimizes its mean square error (MSE) and the error correlation with other ensemble members. To reduce its computational complexity, a new distributed SemiNCL is proposed and experimentally evaluated in this paper.

This paper makes several contributions to the field of ensemble learning and SSL, which are as follows.

1) It proposes a new solution to the effective use of unlabeled data to encourage diversity in an ensemble without sacrificing the accuracy on labeled data. The proposed method does not make strong assumptions about the data distribution, which proves to be beneficial to the performance.
2) The bounds on two penalty coefficients have been derived based on the analysis of the Hessian matrices.
3) We propose a closed-form distributed least square solution to accelerate SemiNCL. The computational and memory complexity have been analyzed, and the empirical evaluation on relatively large data sets has confirmed the accuracy and efficiency of the proposed approach.

NCL [34], [35] has proven to be an effective learning paradigm for classification and regression. However, NCL has only been applied to supervised learning problems, which cannot make use of rich unlabeled data. Moreover, the traditional NCL is optimized by conjugate gradient, which requires additional computational time. In this paper, we address both the problems by introducing the negative correlation term to exploit unlabeled data to promote ensemble diversity

Manuscript received August 11, 2017; revised November 16, 2017; accepted December 12, 2017. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1000905, in part by the National Natural Science Foundation of China under Grant 91546116, Grant 91746209, and Grant 61673363, and in part by the Science and Technology Innovation Committee Foundation of Shenzhen under Grant ZDSYS201703031748284. *(Corresponding author: Xin Yao.)*

H. Chen and B. Jiang are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: hchen@ustc.edu.cn; jiangbb@mail.ustc.edu.cn).

X. Yao is with the Shenzhen Key Laboratory of Computational Intelligence, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China, and also with CERCIA, School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K. (e-mail: xiny@sustc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2784814

without sacrificing the accuracy on labeled data, and using the distributed least square algorithm to accelerate SemiNCL. In particular, the accelerated SemiNCL scales linearly with the total number of labeled and unlabeled samples, and makes a few assumptions about the data distribution, which is more practical for SSL tasks.

The rest of this paper is organized as follows. After the background in Section II, the proposed algorithm is described in Section III. Experimental results and discussions are presented in Section IV. Finally, Section V concludes this paper.

II. Related Work

The idea of considering unlabeled data in supervised training can be traced back to the Shahshahani and Landgrebe's work [44], which showed that the classification performance can be enhanced using unlabeled data. With the rapid development of machine learning, various kinds of SSL algorithms have been developed, e.g., self-learning [41], [43] and generative models [25], [32], co-training [3] and multiview learning [4], [6], graph-based algorithms [1], [28], [49], cluster-based algorithms [31], [33], [42], [45], etc.

The self-learning algorithm [41], [43] uses its own predictions to teach itself. However, the self-learning algorithms usually make "hard" labels for unlabeled data, and the initial misclassifications in self-learning could lead to suboptimal performance. To alleviate this problem, the generative model and expectation-maximization approach have been proposed [25]. Instead of making a "hard" label as in self-learning, the generative model assigns probabilities to labels on unlabeled data, which can be viewed as a kind of "soft" self-training. Still, the initial misclassifications on unlabeled data with soft labels might propagate in the learning process.

The co-training algorithm [3] generates two classifiers using two sufficient and redundant subfeature sets, and then exploits the most confident predictions of each classifier on unlabeled data to "teach" another classifier. Multiview learning [5], [6] generalizes co-training by generating a number of classifiers and let them "teach" each other. The co-training/multiview algorithms are initialized by generating classifiers from *different* input spaces, which are usually implemented by splitting the features into different subsets, i.e., different views. Then these algorithms iteratively retrain on boosted pseudo-labeled sets, based on high-confidence predictions on the unlabeled data. The training follows a greedy agreement-maximization process. The framework is based on two assumptions, namely, the compatibility assumption and the independence assumption. The compatibility assumption imposes that the estimated functions in different views agree on labels in most samples. This assumption will reduce the complexity of the learning problem by only searching over compatible functions. The independence assumption assumes the views to be independent. However, these assumptions, especially the independence assumption, often cannot be satisfied in real-world applications, e.g., when there are no redundant features in the data. In this case, the co-training/multiview learning algorithm may not work well.

The graph-based algorithms directly employ both the labeled and unlabeled data to construct a graph, where the nodes are the labeled and unlabeled samples, and the (possibly weighted) edges reflect "similarity" between the samples. The typical graph-based methods are manifold regularization [1], [27], [28], low-density separation [10], Gaussian fields, and harmonic functions [53]. These methods estimate soft labels, and simultaneously try to impose smoothness on the graph, which can be seen as a smooth function estimator. However, graph-based algorithms make strong assumptions about the data distribution, and suffer from high computational and memory cost since they need to manipulate the graph defined on all labeled and unlabeled data [30], [39].

The cluster-based algorithms try to make the decision boundary pass through the low-density region and simultaneously maximize the margins between different clusters [30]. Some successful semisupervised cluster algorithms, such as TSVM [31], MeanS3VM [33], and ClusterReg [45], follow this paradigm. TSVM uses unlabeled samples to regularize the decision boundary and then seeks the maximum margins. However, the objective function of TSVM is nonconvex, and its performance is sensitive to the initialization [15]. MeanS3VM estimates the label means for all unlabeled samples and then maximizes the margins between these label means. MeanS3VM requires a number of iterations, which limits its efficiencies. ClusterReg uses posterior probabilities generated by a clustering algorithm in its regularization mechanism. When the cluster assumption holds, this algorithm is capable of delivering good performance in the presence of overlapping classes. However, ClusterReg also makes strong assumption about the data distribution, and has to tune more than five parameters. Also, it could generate decision boundaries in the wrong gap between clusters. Such a shortcoming might be avoided with the use of multiple clusters to represent a single class. In this sense, ensemble algorithms can alleviate such an issue by employing several classifiers to overcome potentially incorrect decision boundaries and generate a more robust classifier.

There are several streams of research that generalize the ensemble methods from supervised learning to SSL, for example, ASSEMBLE [2] and semisupervised MarginBoost (MCSSB) [18]. Both the methods work in a greedy manner and maximize the pseudo-margin using boosting method. Recently, SemiBoost [37] has been proposed to use unlabeled samples to improve performance with two regularization terms. UDEED [50], a semisupervised ensemble learning method, proposes to use unlabeled samples to augment the diversity among base learners while maximizing the accuracy of base learners on labeled samples. However, it has difficulty in coping with relatively large-scale data sets due to square computational and memory complexity. RegBoost [14] proposed by Chen and Wang, integrates multiple semisupervised assumptions, including low-density assumption, smoothness assumption, and manifold assumption [11], into the margin cost function and optimized the function using boosting methods. This method achieved comparable classification performance compared with other SSL methods. However, as combining three assumptions, it will take more efforts to optimize several parameters relating to the cost function. If overlapping high-density regions are present, RegBoost

might not establish a good separation between these regions. Moreover, RegBoost requires a square memory complexity, which might be prohibitive for relatively large data sets.

In this paper, we propose a novel semisupervised ensemble learning algorithm, SemiNCL, by generating a group of diverse learners from the *same* input space without making strong assumptions about the data distribution. SemiNCL is based on the implementation of NCL [34], [35], which emphasizes the interaction and cooperation among individual base learners in the ensemble. NCL uses a penalty term to generate biased learners whose errors tend to be negatively correlated. This stabilizes the function estimation in sparsely sampled regions of the input space. NCL has been shown to perform well on a number of applications, including regression [9], [47] and classification problems [29], [36]. It has also been successfully used in learning classifier systems [19]. The theoretical analysis of NCL was conducted by Brown *et al.* [9]. Chen and Yao [12], [13] proposed a regularized NCL (RNCL) for noisy data by including an additional regularization term and solved RNCL by Bayesian inference [12] and multiobjective optimization [13], respectively.

III. Semisupervised Negative Correlation Learning

This section first presents a formulation of negative correlation learning. Then SemiNCL is introduced. An accelerated training algorithm for SemiNCL, as well as a theoretical analysis of the bound for the negative correlation imposing parameter in SemiNCL are given.

A. Negative Correlation Learning

NCL [34], [35] introduces a correlation penalty term into the error function of each individual network in the ensemble, so that all the networks can be trained interactively on the same training set. Given a training set $\{\mathbf{x}_n, y_n\}_{n=1}^N$, NCL combines M neural networks $f_i(\mathbf{x})$ to form an ensemble

$$f_{\text{ens}}(\mathbf{x}_n) = \frac{1}{M}\sum_{i=1}^{M} f_i(\mathbf{x}_n).$$

To train network f_i, the cost function e_i for network i is defined by

$$e_i = \frac{1}{N}\sum_{n=1}^{N}(f_i(\mathbf{x}_n) - y_n)^2 + \lambda\frac{1}{N}p_i \tag{1}$$

where $\lambda \geq 0$ is a weighting parameter on the penalty term p_i

$$\begin{aligned} p_i &= \sum_{n=1}^{N}\left\{(f_i(\mathbf{x}_n) - f_{\text{ens}}(\mathbf{x}_n))\cdot\sum_{j\neq i}(f_j(\mathbf{x}_n) - f_{\text{ens}}(\mathbf{x}_n))\right\} \\ &= -\sum_{n=1}^{N}(f_i(\mathbf{x}_n) - f_{\text{ens}}(\mathbf{x}_n))^2. \end{aligned} \tag{2}$$

The first term on the right-hand side of (1) is the empirical training error of network i. The second term p_i is a correlation penalty function. It is to negatively correlate each network's error with errors for the rest of the ensemble by minimizing p_i. The parameter λ manages the tradeoff between the penalty term and the training error term. When $\lambda = 0$, the individual ensemble member is trained independently. When λ increases, more and more emphasis is placed on minimizing the correlation-based penalty.

B. Formulation of SemiNCL

In the semisupervised setting, the learner is trained on both the labeled sample set $D_l = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ with N labeled examples and the unlabeled sample set $D_u = \{\mathbf{x}_{N+1}, \ldots, \mathbf{x}_{N+V}\}$ with V unlabeled points. Let $D = \{D_l, D_u\}$ represents a training set. The ith individual network f_i in SemiNCL is assumed to be a linear combination of K nonlinear basis functions

$$f_i = \sum_{k=1}^{K} w_{ki}\phi_{ki} = \Phi_i^T\mathbf{w}_i \tag{3}$$

where $\mathbf{w}_i = (w_{1i}, \ldots, w_{Ki})^T$ and $\Phi_i = (\phi_{1i}, \ldots, \phi_{Ki})$ denote the weight vector and basis functions vector in the ith network, respectively.

For instance, multilayer perceptions with linear output nodes, polynomial neural networks, and radial basis functions (RBF) networks [16] are potential estimators in this class. In this paper, we will employ RBF as the base learners.

In order to exploit the unlabeled data in the training process, we include unlabeled data in the error function of f_i

$$\begin{aligned} e_i^{\text{semi}} = \frac{1}{N}\sum_{n=1}^{N}(f_i(\mathbf{x}_n) - y_n)^2 - \frac{\lambda_1}{N}\sum_{n=1}^{N}(f_i(\mathbf{x}_n) - f_{\text{ens}}(\mathbf{x}_n))^2 \\ - \frac{\lambda_2}{V}\sum_{n=N+1}^{N+V}(f_i(\mathbf{x}_n) - f_{\text{ens}}(\mathbf{x}_n))^2. \end{aligned} \tag{4}$$

Comparing this error function with the cost function of NCL (1), SemiNCL employs both the labeled and unlabeled data to calculate the negative correlation term. It has been shown both theoretically and empirically that diversity is beneficial to ensemble learning [8], [46]. By considering both the labeled and unlabeled negative correlation terms in SemiNCL, diversity can be promoted in unlabeled area, and the generalization, which is the tradeoff between accuracy and diversity, could be improved.

Note that no labels are needed for the correlation calculations in the penalty terms of (4). Similar to the original NCL, the requirement of a certain degree of negative correlation among the ensemble members on both the labeled and unlabeled data (controlled by λ_1 and λ_2, respectively) can lead to a more robust and smoother function estimates over the input regions containing samples without labels.

The scaled conjugate gradient (SCG) [38] algorithm can be employed to optimize SemiNCL. According to (4), the minimization of the error function can be achieved by minimizing the error functions of each individual network. In this way, SemiNCL$_{\text{SCG}}$ decomposes the learning task into a number of subtasks for each individual member. However, the computational complexity of gradient training is high due to

the calculation of gradients and objectives values. To reduce the computational complexity, an accelerated SemiNCL is presented in the following section.

C. Accelerated SemiNCL

In this section, we propose a closed-form solution[1] to SemiNCL based on least square optimization and analyze its runtime complexity.

Assume that for each RBF network $i \in \{1, \ldots, M\}$, the matrices $\mathbf{L}_i \in R^{h_i \times N}$ and $\mathbf{U}_i \in R^{h_i \times V}$ represent the network outputs for the labeled and unlabeled data, respectively, where h_i is the number of hidden nodes in the ith RBF network.

For fixed λs, the training of SemiNCL minimizes[2]

$$E = \frac{1}{MN}\sum_{i=1}^{M}\left\|\mathbf{L}_i^T\mathbf{w}_i - \mathbf{y}\right\|^2 - \frac{\lambda_1}{MN}\sum_{i=1}^{M}\left\|\mathbf{L}_i^T\mathbf{w}_i - \mathbf{f}_{ens}^l\right\|^2 - \frac{\lambda_2}{MV}\sum_{i=1}^{M}\left\|\mathbf{U}_i^T\mathbf{w}_i - \mathbf{f}_{\text{ens}}^u\right\|^2 \quad (5)$$

over $\mathbf{w} = (\mathbf{w}_1^T, \ldots, \mathbf{w}_M^T)^T \in R^{h_1} \times \cdots \times R^{h_M}$, where $\mathbf{y} \in R^N$ is the column vector of training labels

$$\mathbf{f}_{\text{ens}}^l = \frac{1}{M}\sum_{i=1}^{M}\mathbf{L}_i^T\mathbf{w}_i$$

is the ensemble output vector for labeled data and

$$\mathbf{f}_{\text{ens}}^u = \frac{1}{M}\sum_{i=1}^{M}\mathbf{U}_i^T\mathbf{w}_i$$

is the output vector of the ensemble for unlabeled data.

The solution to this quadratic optimization problem can be obtained by setting $(\partial E/(\partial \mathbf{w}_i)) = 0$, that is

$$\left(\mathbf{L}_i\mathbf{L}_i^T - \lambda_1\beta\mathbf{L}_i\mathbf{L}_i^T - \frac{N}{V}\beta\lambda_2\mathbf{U}_i\mathbf{U}_i^T\right)\mathbf{w}_i + \sum_{j=1, j\neq i}^{M}\left(\frac{\lambda_1}{M}\mathbf{L}_i\mathbf{L}_j^T + \frac{N}{MV}\lambda_2\mathbf{U}_i\mathbf{U}_j^T\right)\mathbf{w}_j = \mathbf{L}_i^T\mathbf{y} \quad (6)$$

where $\beta = 1 - (1/M)$. Let

$$\mathbf{G}_i = \mathbf{L}_i\mathbf{L}_i^T - \lambda_1\beta\mathbf{L}_i\mathbf{L}_i^T - \frac{N}{V}\beta\lambda_2\mathbf{U}_i\mathbf{U}_i^T$$

and

$$\mathbf{G}_{ij} = \frac{\lambda_1}{M}\mathbf{L}_i\mathbf{L}_j^T + \frac{N}{MV}\lambda_2\mathbf{U}_i\mathbf{U}_j^T, \quad i \neq j.$$

The equation can be written as

$$\begin{pmatrix} \mathbf{G}_1 & \mathbf{G}_{12} & \cdots \\ \mathbf{G}_{21} & \mathbf{G}_2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \cdot \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathbf{L}_1\mathbf{y} \\ \mathbf{L}_2\mathbf{y} \\ \vdots \end{pmatrix}.$$

[1]It does not suffer from local minima as the case for gradient descent. The closed-form solution derived in this section is applicable to RBF networks. The similar derivations could be made to multilayer perceptions with linear output nodes and polynomial neural networks as well.

[2]Boldface and capital letters refer to matrices, and the subscript stands for the indexed number, e.g., $\mathbf{L}_i$ is the output matrix of RBF nodes in the ith ensemble member. Boldface and lower-case letters refer to vectors, and subscript stands for the indexed number, e.g., $\mathbf{w}_i$ is the weight vector of the ith ensemble member.

The solution to $\mathbf{w}$ thus can be obtained by solving the above linear equations.

The complexity of evaluating the $\mathbf{G}$ matrix is $O(M^2H^2(N + V))$ and the complexity of inverting the $\mathbf{G}$ matrix is $O(W^3)$, where H is the average number of weights in $\mathbf{w}_i$, i.e., the average number of hidden nodes in each RBF network, and W ($W = MH$) is the total number of weights in $\mathbf{w}$. This computational complexity $O(W^2(N + V) + W^3)$ is high for large data sets with large ensembles.

To reduce the complexity, the algorithm can be implemented by an iterative, distributed algorithm that shares the predictions of labeled and unlabeled data. Since the gradient $(\partial E/(\partial \mathbf{w}_i))$ (6) can be expressed in the following form:

$$\frac{\partial E}{\partial \mathbf{w}_i} = \left(\mathbf{L}_i\mathbf{L}_i^T - \lambda_1\beta\mathbf{L}_i\mathbf{L}_i^T - \frac{N}{V}\beta\lambda_2\mathbf{U}_i\mathbf{U}_i^T\right)\mathbf{w}_i + \sum_{j=1, j\neq i}^{M}\left(\frac{\lambda_1}{M}\mathbf{L}_i\mathbf{f}_j^l + \frac{N}{MV}\lambda_2\mathbf{U}_i\mathbf{f}_j^u\right) - \mathbf{L}_i\mathbf{y}$$

where $\mathbf{f}_j^l = \mathbf{L}_j^T\mathbf{w}_j$ and $\mathbf{f}_j^u = \mathbf{U}_j^T\mathbf{w}_j$ are the predictions of ensemble member j on the labeled and unlabeled data, respectively, the global minimum can be achieved by setting the block gradient to zero using the block coordinate descent [5], [24] over each member i

$$\mathbf{w}_i = \mathbf{G}_i^{-1} \cdot \left(\mathbf{L}_i\mathbf{y} - \sum_{j=1, j\neq i}^{M}\left(\frac{\lambda_1}{M}\mathbf{L}_i\mathbf{f}_j^l + \frac{N}{MV}\lambda_2\mathbf{U}_i\mathbf{f}_j^u\right)\right).$$

By employing the block coordinate descent algorithm [24], the total complexity including evaluating matrix/vector $\mathbf{G}_i$, $\mathbf{L}_i\mathbf{y} - (\lambda t/M)\sum_{j=1, j\neq i}^{M}(\mathbf{L}_i\mathbf{f}_j^l + \mathbf{U}_i\mathbf{f}_j^u)$ and inverting $\mathbf{G}_i$ can be reduced to $O(W(H + M)(N + V) + WH^2)$. The accelerated $\text{SemiNCL}_{\text{dis}}$ is summarized in Fig. 1.

D. Bound on the Penalty Coefficients

The parameters (λ_1, λ_2) are essential for the generalization performance. The two parameters should be neither negative nor too large. With negative parameters, the learners will be positively correlated; with large positive values, the Hessian matrix $\mathbf{H} = ((\partial^2 e_i)/(\partial \mathbf{w}_i \partial \mathbf{w}_j))$ will be nonpositive definite (non-PD), such that useful gradient information is lost from our original objective function. More specifically, the non-PD Hessian will cause the weight divergence as there will be no minimum to converge to. In that case, it would be difficult to obtain a good generalization. This section will derive the conditions under which the Hessian will be non-PD [9].

A necessary condition to ensure that the Hessian is PD is that all elements on the leading diagonal should be positively valued. Specifically, the diagonal elements of the Hessian matrix can be written as

$$\frac{\partial^2 e_i}{w_{ki}^2} = 2\left[1 - \lambda_1\left(1 - \frac{1}{M}\right)^2 - \lambda_2(1 - \frac{1}{M})^2\right]\phi_{ki}^2 \quad (7)$$

where, for RBF network, w_{ki} and ϕ_{ki} are the combinational weight and the output for the kth basis function in the ith network, respectively.

Given: the number of predictors M, a labeled training set $D_l = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\}$ with N labeled examples and an unlabelled sample set $D_u = \{\mathbf{x}_{N+1}, \cdots, \mathbf{x}_{N+V}\}$ with V unlabeled points, the parameters λ_1, λ_2 and α.

Begin: Construct matrix $\mathbf{L}_i$, $\mathbf{G}_i$ and $\mathbf{U}_i$ $(i = 1, \cdots, M)$ as defined in Section III.C. The predictions of $\mathbf{f}_i^l$ and $\mathbf{f}_i^u$ are initialled to zero vectors. Initialize parameters λ_1, λ_2, α, and $t = N/(N+V)$.

- For each ensemble member from $i = 1, 2, \cdots, M$ do:

$$\begin{aligned} \mathbf{w}_i &= \mathbf{G}_i^{-1} \cdot \left(\mathbf{L}_i \mathbf{y} - \sum_{j=1, j\neq i}^{M} \left(\frac{\lambda_1}{M} \mathbf{L}_i \mathbf{f}_j^l + \frac{N}{MV} \lambda_2 \mathbf{U}_i \mathbf{f}_j^u \right) \right). \\ \mathbf{f}_i^l &= \mathbf{L}_i^T \mathbf{w}_i \text{ and } \mathbf{f}_i^u = \mathbf{U}_i^T \mathbf{w}_i \\ & \quad \text{share } \mathbf{f}_i^l \text{ and } \mathbf{f}_i^u \end{aligned}$$

- Repeat for a desired number of iterations or until convergence.

Fig. 1. SemiNCL implemented by the block coordinate descent algorithm $\text{SemiNCL}_{\text{dis}}$.

The entire Hessian matrix is guaranteed to be non-PD when (7) is nonpositive, i.e., the following inequality holds:

$$\frac{\partial^2 e_i}{w_{ki}^2} \leq 0$$

$$\lambda_1 + \lambda_2 \geq \left(\frac{M}{M-1} \right)^2.$$

When λ_1 or λ_2 is varied beyond this lower bound, the Hessian matrix is guaranteed to be non-PD. In other words, this result leads to a upper bound for PD Hessian matrix. When the size of ensemble M increases, the upper bound converges to 1.

Note that the bound is a conservative one since there is a probability that the leading diagonal is all positive, yet the entire matrix is still non-PD. It could be possible to define a tighter bound. However, the advantage of this bound is that it is independent of any network parameters except for the size of ensemble.

IV. Experimental Studies

This section presents experimental results of SemiNCL. First, experimental results of SemiNCL on synthetic data sets are presented to help understand its mechanics. Then, extensive experiments on UCI data sets are carried out to compare SemiNCL with other semisupervised and supervised algorithms, on regression and classification tasks, respectively. After that, statistical analyses are reported, demonstrating the competitiveness of SemiNCL. Finally, we evaluate the scalability of the compared algorithms using several relatively large data sets and analyze the computational and memory complexity of different algorithms.

A. Experimental Settings

In our experiments, RBF networks are employed as individual ensemble members in SemiNCL. We use [22] to initially select the basis functions for RBF networks.[3] The training of RBF network can be separated into two steps. In the first step, the RBF centers are initialized with randomly selected data points from the training data and the kernel widths are determined as the Euclidean distance between each centers and its closest centers. Then, in the second step, we perform gradient descent to tune the centers and widths according to the regularized error function [13], [16], [26]. The maximum number of RBF centers is restricted to 200 for SemiNCL. We use 25 RBF networks to generate the ensemble of SemiNCL.

The parameters (λ_1, λ_2) in SemiNCL are optimized by fivefold cross validation grid search. The search range for both parameters is within $\{0, 0.1, \ldots, 0.5\}$. As described in Section III-D, the bound of $\lambda_1 + \lambda_2 < (M/M-1)^2 = 1.09$ ($M = 25$ in this paper). However, the bound of these parameters is a conservative one and the Hessian matrix might become non-PD even within the bound. In this case, we use a tighter bound $\lambda_1 + \lambda_2 < 1$ instead of $\lambda_1 + \lambda_2 < (M/M-1)^2$ to avoid the unstable situation. Following [36], the number of hidden nodes in the ith RBF network, h_i, can be specified by the users. In our experiments, the value of h_i is randomly selected but restricted in the range of 4–10 [13], and thus, the average number of hidden nodes H is about 7.

B. Synthetic Data Set

In order to facilitate the understanding of SemiNCL, we use a synthetic data set for regression to illustrate the effect of unlabeled data in Fig. 2.

The synthetic regression data are generated by sinc function $\sin(x)/x$ within $[-4\pi, \ldots, 4\pi]$ with Gaussian white noise standard derivations 0.2 and in order to illustrate the effect of unlabeled data using SemiNCL, we have removed some labeled data points within $[\pi, \ldots, 3.5\pi]$ in the right-hand tail of the sinc function. A total of 400 labeled training points are generated by uniformly sampling within $[-4\pi, \ldots, \pi, 3.5\pi, \ldots, 4\pi]$, while 2000 unlabeled points are generated by uniformly sampling within $[\pi, \ldots, 3.5\pi]$. In addition, 4000 test points are generated in the range $[-4\pi, \ldots, 4\pi]$.

We also showed two decomposition terms, i.e., the average individual accuracy, measured by the training error $[\sum_{i=1}^{M} (f_i(\mathbf{x}_n) - y_n)^2$, pentagram] and the sum of labeled and unlabeled diversity (triangle)[4] among ensemble members.

[3]The source code for basis function selection can be downloaded from Yaki Engel's Homepage: http://visl.technion.ac.il/~yaki/c_sources/krls.cc.

[4]In order to show the two terms clearly in Fig. 2, we omit the average coefficient $(1/M)$ in both accuracy and diversity calculation.

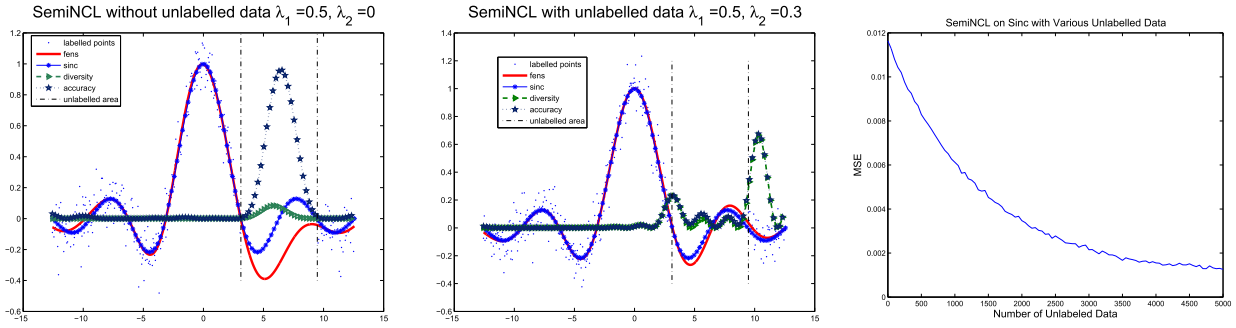


Fig. 2. Illustrations of SemiNCL (red solid line) on sinc data set. (a) SemiNCL without unlabelled data. (b) SemiNCL with 2000 unlabelled points. (c) SemiNCL with different unlabelled data. Some labeled data points are removed. The (c) presents the performance of SemiNCL with various numbers of unlabelled data on this sinc data set.

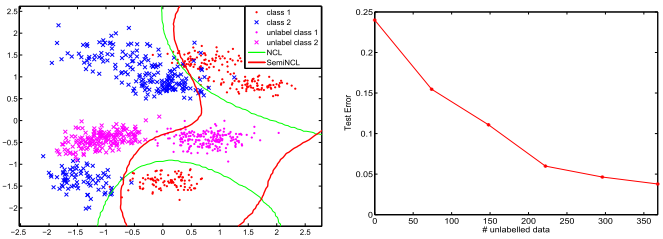


Fig. 3. Illustrations of SemiNCL on the classification synthetic data set. SemiNCL can make use of unlabeled data to generalize much better than NCL (which only considers labeled data) is shown. (a) SemiNCL versus NCL. (b) Test error of SemiNCL.

Fig. 2(a) reports the result of SemiNCL using only labeled data of the sinc function, i.e., NCL. Not surprisingly, the training error is quite large in the unlabeled area. As diversity for unlabeled data is not encouraged in this figure (λ_2 is set to zero), the total diversity is significantly lower than the average individual accuracy. Therefore, the ensemble error (5), i.e., the difference between accuracy and diversity, is largely biased.

Fig. 2(b) shows the result of SemiNCL incorporating 2000 unlabeled data points. The predictions of SemiNCL in the right-hand tail are quite accurate, compared with Fig. 2(a). Although the average individual training error is large in unlabeled area and in the boundary of labeled and unlabeled area,[5] the diversity is well managed to compensate the increased training error due to the negative correlation term. As a result, the ensemble error (5) is smaller.

In Fig. 2(c), we present the MSE of SemiNCL with various numbers of unlabeled data on this sinc data set. As the amount of unlabeled data increases, the performance improves.[6] This is an intuitive example to illustrate that SemiNCL uses both the labeled and unlabeled data to encourage diversity and thus improves the generalization performance.

[5]The accuracy and diversity often change before reaching the boundary of labeled and unlabeled area. When SemiNCL enters the labeled area from unlabeled area, the increased tendency of accuracy/diversity needs a buffer to change to the decreased tendency (refer to the high diversity peak in right-hand side of sinc function) due to smooth regularization.

[6]The parameters are chosen without optimization. Better performance could be achieved with a careful selection of these two parameters.

TABLE I
SUMMARY OF 14 REGRESSION DATA SETS

Name	autoMPG	no2	stock	quake	strike	census	sarcos
Size	398	500	950	2178	625	20459	48933
#Attributes	7	7	9	3	6	119	27
Name	socmob	delta	housing	pollen	concrete	kin40k	adult
Size	1156	9517	506	3848	1030	40000	45222
#Attributes	5	6	13	5	8	8	14

In Fig. 3, we present the performance of SemiNCL with (red) and without (green) unlabeled data on one classification synthetic data set. This synthetic data is generated by eight Gaussians.[7] The first four Gaussians belong to one class and the other four for the other class. The fourth and eighth Gaussians are treated as unlabeled, and the others are treated as labeled. We sample 500 points from each Gaussian. Of them, 184 is used as training or unlabeled points. The left 316 is employed as test points. Therefore, we have a data set with 1104 training, 368 unlabeled, and 2528 test points.

In Fig. 3(a), the decision boundaries of SemiNCL and NCL are illustrated. It is obvious to observe that unlabeled data regularizes the decision boundary in the unlabeled area. Hence, SemiNCL manages to generate a better decision boundary than NCL. Fig. 3(b) illustrates the test error versus the number of unlabeled data points. The test error is reduced by considering more unlabeled points. With more and more unlabeled data being included, SemiNCL seems to converge.

C. UCI Data Sets

In this section, extensive experiments on UCI data sets are presented. We compare SemiNCL with some state-of-the-art semisupervised and supervised learning algorithms for regression and classification, respectively.

1) Regression Problems: Test errors (Mean $\pm$ standard deviation %) of SemiNCL and other semisupervised/supervised algorithms with 5%, 10%, and 20% of

[7]Those centers are $(2, 12)$, $(3, 10)$, $(1, 2)$, $(1.5, 5.5)$, $(4, 12)$, $(5, 10)$, $(3, 2)$, $(4, 5.5)$, respectively. The covariance matrixes in MATLAB notations are $[0.625, -0.2165; -0.2165, 0.875]$, $[0.25, 0; 0, 0.25]$, $[0.2241, -0.1368; -0.1368, 0.9759]$, $[0.2375, 0.1516; 0.15160.4125]$, $[0.25, 0; 0, 0.25]$,$[0.25, 0; 0, 0.25]$, $[0.25, 0; 0, 0.25]$, $[0.25, 0; 0, 0.25]$, respectively.

labeled data. The best result for each data set is illustrated in boldface.

This paper uses 14 real-world regression data sets from UCI [40], DELVE,[8] Statlib,[9] and other sources[10] in the experimental study. In the *census* data set, the data points with missing values are removed to facilitate the following processing. As in [17], we convert the classification problem, *adult*, to a regression problem by predicting $+1$ for examples in class 1 and -1 for the other class. The characteristics of these data sets are summarized in Table I. All data sets were input-normalized dimension-wise to have zero mean and unit standard deviation.

In the following experiments, we randomly partitioned each data set into labeled/unlabeled/test data sets according to different ratios. Specifically, 75% of the data are used as the labeled/unlabeled training sets, whereas the remaining 25% of the data are left as test set. For each training set, we partition the training set under different label rates. For example, assuming that there are 1000 examples in the training set, if the label rate is 10%, 100 examples with their labels are selected for labeled sets, whereas the remaining 900 examples are taken as unlabeled training set without labels.

In the experiments, we follow the methodology of [51] and run each algorithm on each data set for 100 times with random split of the labeled/unlabeled/test sets for each data set and report the average MSE.

In order to further evaluate the performance of SemiNCL, we compare SemiNCL with the three existing semisupervised regression methods, namely, co-training regressors (COREG) [51], manifold regularization for regression [1], and co-regularized least squares regression (CoLSR) [5]. We use these algorithms for benchmark comparisons as they are typical representatives for semisupervised regression algorithms.

The number of nearest neighbors k for COREG [51] and CoLSR [1] are selected by fivefold cross validation grid search over $k \in \{1, 3, \ldots, 30\}$. The source code of COREG can be obtained online.[11] In manifold regularization [1], the Gaussian kernel is used to construct the nearest neighbor set with the kernel parameter setting to $\sigma = 1/(N+V)^2 \sum_{i,j=1}^{N+V} \|x_i - x_j\|^2$. The number of nearest neighbors k and the other two parameters, γ_A and γ_I, are selected by fivefold cross validation grid search, where γ_A and γ_I correspond to the RKHS norm regularization parameter and manifold regularization parameter, respectively. The search ranges $k \in \{1, \ldots, 10\}$ and $\gamma_A, \gamma_I \in \{10^{-2}, \ldots, 1\}$ are adopted. In CoLSR, we follow the parameter setup in [5] and use a Gaussian kernel $k(x_i, x_j) = \exp(-\|x_i - x_j\|^2/\sigma)$ with $\sigma = 1/(N+V)^2 \sum_{i,j=1}^{N+V} \|x_i - x_j\|^2$. The regularization parameter ν is selected by cross validation within the search range $\nu \in \{0, 0.1, \ldots, 1\}$. Note that σ and ν depend only on the labeled examples. In the case of multiple views, σ_v and ν_v are computed from the attributes in the respective view v.

[8]http://www.cs.toronto.edu/ delve/data/data sets.html

[9]http://lib.stat.cmu.edu/data sets/

[10]http://ida.first.fraunhofer.de/ anton/data/, http://www.gaussianprocess.org/gpml/data/

[11]http://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/annex/COREG.htm

In order to compare our proposed algorithm with existing supervised learning algorithms, we also examine the performance of the three supervised learning algorithms: random forests (RF) [7], NCL [34], and a single RBF network [13], [16]. In RF, 100 classification and regression trees have been generated to construct the forests. In NCL, we use the same ensemble size $M = 25$ and the parameter λ is selected by fivefold cross validation from $\{0, 0.1, \ldots, 1\}$. The parameters (λ_1, λ_2) in SemiNCL are optimized by fivefold cross validation grid search. The search range for both parameters is within $\{0, 0.1, \ldots, 0.5\}$.

Table II reports the results of these algorithms under different label rates. The lowest MSE among the compared algorithms, under each label rate have been boldfaced. First, let us consider the comparison between SemiNCL, Manifold, COREG, and CoLSR. Table II shows that SemiNCL achieves 41 wins out of a total of 56 comparisons against all the other compared algorithms. In contrast, manifold, COREG, and CoLSR win in 1, 10, and 3 cases, respectively.[12] COREG performs better when the label rate increases, which means that fewer misclassification propagates in the co-training stage.

Second, the result in Table II indicates that SemiNCL achieves highly competitive performance in comparison with the supervised learning algorithms. For example, SemiNCL always wins against the RBF network on all data sets; SemiNCL outperforms NCL except on the census data set where they both achieve the same performance. RF only win twice and tie twice with SemiNCL among 56 comparisons.

We also notice that with the increased label rate, the difference between SemiNCL and NCL seems to decrease. Recall that SemiNCL considers both the labeled and unlabeled correlation terms, whereas NCL only considers the labeled correlation term. When the label rate increases, fewer unlabeled data are presented and the unlabeled correlation term plays a less important role, which causes the difference between SemiNCL and NCL to decrease.

2) Classification Problems: Here, we select 12 classification data sets from the UCI machine learning repository [40]. The characteristics of each data set are reported in Table III.

In the experiments, we follow the setting as in [14] and run each algorithm on each data set for 20 times with random split of the labeled/unlabeled/test instances for each data set, in which 20% instances for testing. The average classification error on test sets with different label ratios 5%, 10%, and 20%, are reported. The true class structure and the corresponding assumption in semisupervised classification (SSC) are unknown for these real-world data sets.

The proposed algorithm is compared with several state-of-the-art semisuperivised algorithms that use different assumptions: Manifold [1] and SS-ELM [28] with manifold assumption; MeanS3VM [33] and ClusterReg [45] based on

[12]Since manifold learning and CoRLSR algorithms need to calculate the kernel matrix using all labeled and unlabeled data, they could not be applied to four relatively large data sets due to large memory requirements in our computational environment. Therefore, we use the symbol "-" to indicate that the memory is insufficient for the calculation. The JAMA matrix package (http://math.nist.gov/javanumerics/jama/) used in the COREG algorithm cannot invert the singular matrix generated by the census data set, and we denote this as "-,"too.

TABLE II

MSE AND STANDARD DEVIATION (CLASSIFICATION ERROR FOR ADULT DATA SET) OF SEMINCL AND OTHER SEMISUPERVISED/ SUPERVISED ALGORITHMS UNDER DIFFERENT LABEL RATIOS ON 14 REAL-WORLD DATA SETS. THE BEST RESULT FOR EACH DATA SET IS ILLUSTRATED IN BOLDFACE

Label Rate	Data sets	SemiNCL	Manifold	COREG	CoRLSR	RF	NCL	RBF
0.1	autoMpg	**0.187±0.050**	0.232±0.055	0.214±0.071	0.220±0.063	0.224±0.047	0.240±0.048	0.252±0.054
	no2	**0.938±0.082**	1.019±0.084	1.103±0.091	1.003±0.074	1.020±0.111	1.043±0.083	1.076±0.135
	stock	**0.049±0.010**	0.062±0.012	0.053±0.008	0.060±0.010	0.071±0.013	0.056±0.013	0.073±0.021
	quake	**0.915±0.074**	1.042±0.083	1.096±0.081	0.999±0.071	1.107±0.088	1.078±0.098	1.114±0.109
	strike	**0.923±0.441**	0.965±0.529	1.086±0.492	0.957±0.458	1.032±0.517	1.054±0.531	1.220±0.635
	socmob	**0.407±0.167**	0.461±0.186	0.435±0.155	0.441±0.180	0.499±0.196	0.448±0.186	0.617±0.286
	delta	**0.359±0.016**	0.411±0.022	0.405±0.021	0.391±0.019	0.392±0.020	0.377±0.016	0.471±0.023
	housing	0.362±0.090	0.391±0.097	0.389±0.086	0.390±0.107	**0.345±0.095**	0.372±0.090	0.375±0.081
	pollen	1.023±0.020	1.048±0.027	1.029±0.024	**1.009±0.023**	1.077±0.025	1.061±0.022	1.063±0.030
	concrete	**0.318±0.048**	0.362±0.048	0.347±0.045	0.387±0.042	0.381±0.042	0.331±0.042	0.369±0.039
	census	**0.007±0.002**	-	-	-	0.008±0.002	0.009±0.003	0.013±0.002
	sarcos	0.015±0.001	-	**0.014±0.001**	-	0.016±0.001	0.016±0.001	0.021±0.002
	kin40k	**0.026±0.006**	-	0.029±0.007	-	0.028±0.007	0.029±0.006	0.033±0.012
	adult	**0.147±0.004**	-	0.171±0.004	-	0.154±0.003	0.161±0.004	0.199±0.008
0.3	autoMpg	**0.150±0.036**	0.209±0.063	0.181±0.051	0.195±0.053	0.164±0.042	0.168±0.036	0.169±0.038
	no2	0.936±0.079	0.995±0.064	**0.916±0.092**	0.971±0.081	1.020±0.101	1.031±0.079	0.945±0.063
	stock	**0.024±0.008**	0.033±0.009	0.027±0.005	0.037±0.009	0.035±0.007	0.038±0.010	0.049±0.019
	quake	**0.906±0.062**	1.016±0.078	1.025±0.078	1.001±0.075	1.092±0.072	1.011±0.073	1.029±0.069
	strike	0.846±0.508	0.857±0.502	0.862±0.515	**0.837±0.429**	0.915±0.547	0.950±0.528	1.072±0.444
	socmob	0.356±0.116	0.351±0.105	**0.339±0.093**	0.410±0.119	0.415±0.164	0.397±0.137	0.463±0.287
	delta	**0.353±0.013**	0.379±0.021	0.369±0.017	0.372±0.017	0.382±0.016	0.369±0.014	0.463±0.021
	housing	**0.205±0.062**	0.236±0.081	0.219±0.063	0.223±0.066	0.219±0.067	0.215±0.050	0.242±0.080
	pollen	**0.914±0.020**	1.047±0.028	1.023±0.028	1.008±0.024	1.072±0.026	1.019±0.020	1.030±0.029
	concrete	**0.211±0.028**	0.255±0.032	0.232±0.027	0.246±0.025	0.212±0.024	0.231±0.023	0.268±0.030
	census	**0.004±0.002**	-	-	-	0.005±0.001	0.006±0.003	0.009±0.001
	sarcos	**0.009±0.000**	-	**0.009±0.000**	-	0.010±0.000	0.010±0.000	0.013±0.001
	kin40k	0.021±0.005	-	**0.020±0.007**	-	0.022±0.005	0.023±0.004	0.027±0.010
	adult	**0.145±0.004**	-	0.168±0.006	-	0.146±0.004	0.161±0.004	0.192±0.006
0.5	autoMpg	**0.142±0.035**	0.179±0.054	0.163±0.051	0.163±0.035	0.153±0.041	0.151±0.037	0.163±0.029
	no2	**0.849±0.041**	0.922±0.045	0.938±0.042	0.927±0.046	0.911±0.046	0.927±0.066	0.998±0.063
	stock	0.021±0.005	0.028±0.006	**0.020±0.004**	0.026±0.006	0.025±0.005	0.029±0.005	0.043±0.016
	quake	**0.882±0.059**	1.015±0.079	1.014±0.084	0.969±0.072	1.050±0.072	0.982±0.075	1.010±0.078
	strike	**0.818±0.380**	0.820±0.495	0.841±0.496	0.824±0.519	0.896±0.482	0.915±0.418	0.954±0.396
	socmob	0.349±0.101	**0.348±0.102**	0.354±0.089	0.374±0.095	0.382±0.138	0.388±0.122	0.414±0.258
	delta	**0.351±0.013**	0.371±0.019	0.359±0.018	0.366±0.014	0.375±0.015	0.367±0.013	0.457±0.019
	housing	0.190±0.063	0.199±0.083	**0.187±0.072**	0.193±0.071	0.191±0.062	0.197±0.062	0.238±0.073
	pollen	**0.909±0.021**	1.045±0.029	1.021±0.031	1.010±0.024	1.069±0.025	1.014±0.024	1.018±0.028
	concrete	**0.153±0.024**	0.193±0.031	0.215±0.026	0.213±0.026	0.167±0.024	0.198±0.022	0.224±0.028
	census	**0.004±0.001**	-	-	-	**0.004±0.001**	**0.004±0.002**	0.009±0.002
	sarcos	**0.006±0.000**	-	0.007±0.000	-	0.008±0.000	0.008±0.000	0.010±0.001
	kin40k	**0.016±0.003**	-	0.018±0.004	-	0.019±0.003	0.020±0.004	0.021±0.008
	adult	**0.145±0.003**	-	0.152±0.004	-	0.146±0.003	0.161±0.003	0.189±0.003
0.7	autoMpg	**0.120±0.024**	0.157±0.042	0.149±0.035	0.153±0.039	0.134±0.023	0.128±0.019	0.142±0.036
	no2	**0.822±0.035**	0.910±0.043	0.852±0.041	0.849±0.039	0.856±0.048	0.885±0.057	0.907±0.045
	stock	0.017±0.004	0.026±0.006	**0.016±0.003**	0.024±0.005	0.020±0.003	0.027±0.004	0.030±0.013
	quake	**0.876±0.052**	0.989±0.077	1.012±0.081	0.962±0.071	1.043±0.077	0.978±0.070	1.004±0.074
	strike	0.811±0.341	0.809±0.398	0.818±0.374	**0.794±0.371**	0.796±0.339	0.874±0.389	0.939±0.326
	socmob	0.340±0.084	0.342±0.092	**0.336±0.078**	0.371±0.092	0.381±0.134	0.365±0.118	0.404±0.214
	delta	**0.347±0.012**	0.368±0.015	0.356±0.015	0.364±0.013	0.370±0.014	0.368±0.012	0.431±0.018
	housing	0.153±0.059	0.186±0.078	0.163±0.071	0.164±0.072	**0.143±0.053**	0.170±0.060	0.207±0.059
	pollen	**0.906±0.021**	1.043±0.023	1.021±0.028	1.003±0.023	1.065±0.023	1.013±0.024	1.018±0.028
	concrete	**0.126±0.022**	0.146±0.031	0.145±0.024	0.147±0.031	0.139±0.015	0.150±0.020	0.181±0.026
	census	**0.003±0.001**	-	-	-	**0.003±0.001**	**0.003±0.002**	0.009±0.001
	sarcos	**0.005±0.000**	-	0.006±0.000	-	0.007±0.000	0.006±0.000	0.008±0.000
	kin40k	0.013±0.001	-	**0.012±0.001**	-	0.016±0.001	0.016±0.001	0.019±0.009
	adult	**0.142±0.003**	-	0.149±0.003	-	0.145±0.003	0.162±0.003	0.182±0.003

The results are averaged over 100 runs. Since manifold learning and CoRLSR have to calculate the kernel matrix using all labeled and unlabeled data and thus could not be applicable to the four relatively large data sets (i.e., census, sarcos,kin40k and adult). Therefore, we use "-" to indicate that the memory is not enough for the calculation. The JAMA matrix package used in COREG algorithm cannot invert the singular matrix generated by census data set, and we denote this as "-", too.

cluster assumption; three successful semisupervised ensemble classifiers, i.e., SemiBoost [37], UDEED [50], and RegBoost [14], which based on multiple SSC assumptions.

Two supervised ensemble learning algorithms, AdaBoost [23] and NCL [34] are used as baselines. NCL uses the same ensemble size $M = 25$ and the parameter λ is selected by fivefold cross validation, where λ is selected from $\{0, 0.1, \ldots, 1\}$. AdaBoost uses the default settings and the number of weak learners is 25.

The parameter settings for Manifold are same as that for regression problems, and $\gamma_A, \gamma_I \in \{10^{-2}, \ldots, 10^2\}$ are adopted. In SS-ELM, the size of hidden neurons is fixed as 100, and other parameters are tuned by adopting the same settings as in Manifold. The parameter settings of MeanS3VM follow the setup in [33]. The linear kernel is used and the regularization parameters $c_1, c_2 \in \{10^{-2}, \cdots, 10^2\}$. These parameters are selected by fivefold cross validation grid search. ClusterReg [45] uses the partition information of a

TABLE III
SUMMARY OF DATA SETS FOR CLASSIFICATION PROBLEMS

Name	Australian	BUPA	German	Harberman	Horse	House
Size	690	345	1000	306	368	435
#Attributes	14	6	24	3	27	16
Name	Ionosphere	Mass	Pima	SPECT	Transfusion	WDBC
Size	351	961	768	267	748	569
#Attributes	34	5	8	22	4	30

clustering algorithm to regularize the decision boundary of a classifier. The clustering algorithm in ClusterReg is chosen from K-means, Gaussian mixture models, self-tuning spectral clustering, and Fuzzy Gustafson Kessel; the number of clusters was set as 1, 2, or 3 times the number of classes; the number of neighbors is picked from {1; 10; 30}; and the parameter that controls the steepness of the mapping from similarity to penalization $\kappa = \{2; 5; 9; 12\}$. The tradeoff parameter λ is fixed at 0.2, the number of hidden nodes and epochs are 15 and 50, respectively. In SemiBoost, we follow [37] and the number of iterations T is set to 20. UDEED [50] employs unlabeled examples to measure the diversity. In UDEED, the ensemble size is 25 and the cost parameter C is adjusted by fivefold cross validation from $\{10^{-2}, \ldots, 10^{2}\}$. In RegBoost, we follow [14] and perform grid search for the best combination of parameters. The number of neighbors is within $\{3, 4, 5, 6\}$. The resampling rate in the iterations is in $\{0.1, 0.25, 0.5\}$.

Table IV reports the generalization error for the employed algorithms on the presence of 5%, 10%, and 20% of labeled data, respectively.

When compared to the state-of-the-art semisupervised ensemble methods, i.e., SemiBoost, UDEED, and RegBoost, SemiNCL is able to deliver better results under all amounts of labeled data. SemiNCL shows significantly better results against the supervised method AdaBoost and NCL with 5%, 10%, and 20% of labeled data. This fact indicates that SemiNCL can effectively employ the unlabeled data to promote the performance.

Then, let us compare SemiNCL with other SSL algorithms. Table IV shows that SemiNCL achieves 17 wins against all the other compared algorithms in the total of 36 comparisons. In contrast, MeanS3VM wins over all other algorithms on three experiments, Manifold wins on four experiments, and SS-ELM wins on six experiments in a total of 36 comparisons. SS-ELM performs better when the label rate increases, which means that the graph-based algorithm used in SS-ELM makes fewer misclassification with more labeled data.

According to the experimental results for regression (Table II) and classification (Table IV), some SSL algorithms, such as Manifold, CoRLSR for regression, and ClusterReg, RegBoost for classification, can be slightly worse than supervised learning algorithms on some data sets, especially with small label rates. However, such results do not mean that SSL does not help. This is because different SSL algorithms rely on different assumptions and when these assumptions are not satisfied, these semisupervised algorithms will not work well. Compared with these SSL algorithms, AdaBoost and NCL are strong supervised algorithms with few assumptions on data.

SemiNCL utilizes unlabeled data to encourage diversity within ensemble members, and the underlying assumption[13] is relatively weaker than that of the other semisupervised algorithms, such as Manifold and ClusterReg. Therefore, SemiNCL can achieve better performance than the supervised algorithms on a wide range of different data sets.

The experimental results reveal that although using the unlabeled data helps the generalization of SemiNCL, the large improvement often occurs when dealing with relatively few labeled and large unlabeled data.

D. Statistical Comparisons Over Multiple Data Sets

To compare the performance of the proposed method, we perform Friedman tests [20], which is based on the ranks of compared methods to compare multiple classifiers on multiple data sets. The Friedman tests together with the Bonferroni-Dunn test [21] were used as post-hoc tests when all estimators are compared with the control estimator. The performance of pairwise comparison is significantly different if the corresponding average ranks[14] differ by at least the critical difference

$$\mathrm{CD} = q_\alpha \sqrt{\frac{j(j+1)}{6T}} \tag{8}$$

where j is the number of algorithms, T is the number of data sets, and critical values q_α can be found in [20]. For example, when $j = 5$, $q_{0.05} = 2.498$, where the subscript 0.05 is the significance level.

In this significant test, we would like to choose SemiNCL as the control classifier to be compared with. For semisupervised regression tasks, COREG [51], Manifold [1], CoLSR [5], and NCL [34] will be compared with the control classifier SemiNCL. Since Manifold and CoLSR could not deal with relatively large data sets, we only consider the first 10 data sets in the statistical study. For SSC tasks, UDEED [50], Manifold [1], ClusterReg [45], and NCL [34] will be chosen. We include NCL in this comparison since SemiNCL is an extension of NCL.

Figs. 4 and 5 report the Friedman test results for regression and classification, respectively. Since we employ the significance level 0.05, the critical difference for regression is $\mathrm{CD} = 1.77$ with $j = 5$ and $T = 10$, and the critical difference for classification is $\mathrm{CD} = 1.61$ with $j = 5$ and $T = 12$. Several observations can be made from our results.

For regression tasks, the differences of SemiNCL versus Manifold, SemiNCL versus CoLSR (except when the label rate is 0.7), and SemiNCL versus NCL are greater than the critical difference, so the differences are significant. It means that SemiNCL is significantly better than Manifold, CoLSR (except when label rate is 0.7), and NCL in these cases.

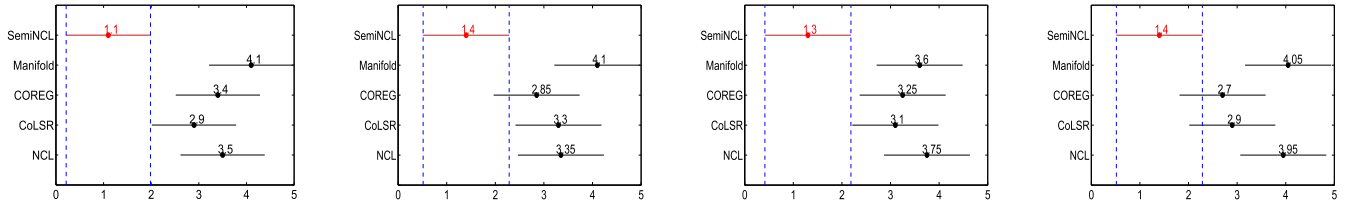
[13]To encourage the diversity using unlabeled data, the unlabeled data should follow the similar distribution/tendency of the labeled data, which is a kind of "smoothness" in the data space. This is the underlying assumption of SemiNCL.

[14]We rank these algorithms and record the ranking of each algorithm as 1, 2 and so on. Average ranks are assigned in case of ties. The average rank of a single algorithm is obtained by averaging over all of data sets. Please refer to Tables II and IV for the mean rank of these algorithms.

TABLE IV

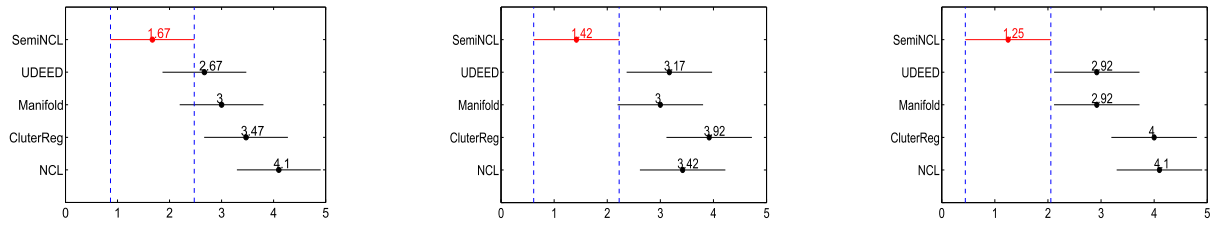
TEST ERRORS (MEAN ± STANDARD DEVIATION %) OF SEMINCL AND OTHER SEMISUPERVISED/SUPERVISED ALGORITHMS WITH 5%, 10%, AND 20% OF LABELED DATA ON 12 UCI DATA SETS. THE BEST RESULT FOR EACH DATA SET IS ILLUSTRATED IN BOLDFACE

	Results for 5% of labeled data									
Datasets	AdaBoost	NCL	MeanS3VM	ClusterReg	Manifold	SS-ELM	UDEED	SemiBoost	RegBoost	*SemiNCL*
Australian	21.23 ± 6.67	26.65 ±2.32	16.92±4.09	29.96 ±12.93	**14.93 ±3.20**	15.69 ±3.11	16.56 ±4.28	18.67±5.10	17.81 ±5.34	15.87 ±3.73
BUPA	43.07±7.60	45.57±2.33	42.97 ±9.16	43.70 ±8.28	45.14 ±10.85	43.55 ±9.32	43.19 ±8.30	44.67 ±8.28	43.55 ±7.92	**41.30 ±8.69**
German	31.85 ±2.96	36.28 ±1.59	30.75 ±3.92	30.05 ±3.15	35.87 ±4.18	33.50 ±3.71	30.32 ±3.28	30.34 ±3.05	30.71 ±4.27	**29.33 ±3.52**
Haberman	33.69 ±10.14	32.54 ±2.90	25.98 ±7.37	27.30 ±11.25	28.69 ±13.08	33.69 ±11.56	**25.49 ±5.25**	27.45 ±10.58	31.37 ±7.29	26.43 ±7.94
Horse	38.09 ±7.41	35.21 ±2.25	38.18 ±7.98	36.62 ±9.88	39.80 ±7.18	39.26 ±6.87	37.03 ±6.08	38.14 ±7.07	**33.43 ±4.47**	36.59 ±6.30
House	23.97 ±8.67	19.43 ±2.24	10.46 ±4.25	18.39 ±9.32	7.87 ±2.77	7.36 ±3.19	7.76 ±2.53	**5.82 ±3.05**	10.81 ±4.47	7.59 ±2.50
Iono	23.86 ±9.05	18.11 ±2.18	22.00 ±7.39	33.36 ±3.54	23.26 ±9.61	21.71 ±7.51	25.79 ±7.22	21.21 ±7.23	**18.05 ±7.76**	18.71 ±5.29
Mass	21.30 ±3.67	25.77 ±2.25	21.83 ±3.38	21.48 ±3.13	21.22 ±2.80	**21.17 ±2.38**	21.72 ±3.01	23.30 ±6.28	25.42 ±4.94	25.12 ±8.93
Pima	31.56 ±6.57	31.88 ±2.06	**27.44 ±4.44**	34.25 ±3.83	28.77 ±5.72	29.48 ±4.79	30.06 ±4.29	34.52 ±3.30	31.72 ±4.48	31.04 ±3.76
SPECT	31.13 ±8.84	20.28 ±5.91	20.94 ±6.09	19.45±4.79	25.19 ±12.25	23.68 ±6.26	19.53 ±5.88	30.81 ±9.31	20.70 ±8.59	**18.92 ±4.84**
Transfusion	26.63 ±3.98	36.87 ±2.82	24.17 ±2.73	24.17 ±2.69	30.17 ±5.87	33.13 ±4.79	24.00 ±2.87	24.34 ±2.72	24.61 ±6.46	**23.55 ±2.79**
WDBC	24.39 ±5.91	17.43 ±2.21	8.60 ±4.70	13.16 ±5.79	**4.96 ±1.98**	5.48 ±1.68	6.49 ±2.92	12.11 ±2.98	6.85 ±3.77	6.67 ±3.79
	Results for 10% of labeled data									
Datasets	AdaBoost	NCL	MeanS3VM	ClusterReg	Manifold	SS-ELM	UDEED	SemiBoost	RegBoost	*SemiNCL*
Australian	16.52 ±3.32	27.17 ±2.03	14.48 ±2.52	22.14 ±10.08	**14.17 ±2.46**	14.31 ±2.53	14.28 ±2.70	15.57 ±2.54	17.58 ±3.54	14.68 ±2.84
BUPA	40.20 ±7.25	40.07 ±2.32	39.71 ±6.69	44.13 ±9.11	43.41 ±5.79	41.09 ±7.67	43.48 ±9.66	41.71 ±7.00	40.71 ±5.41	**39.42 ±5.83**
German	28.67 ±2.61	35.10 ±1.57	28.22 ±2.77	30.05 ±3.15	32.75±4.65	33.30 ±4.51	27.58 ±2.93	29.45 ±2.87	29.50 ±4.25	**27.44 ±2.66**
Haberman	30.33 ±7.26	32.34 ±2.52	24.75 ±4.96	24.59 ±5.10	31.80 ±8.78	37.05 ±8.94	25.43 ±5.32	26.34 ±5.44	30.44 ±7.64	**24.51 ±5.30**
Horse	36.74 ±6.86	35.31 ±2.32	35.47 ±5.24	34.26 ±4.75	34.59 ±5.47	35.68 ±6.22	34.93 ±6.05	34.28 ±4.91	35.04 ±5.65	**33.82 ±4.97**
House	10.63 ±7.41	19.45 ±2.46	9.31 ±4.03	18.45 ±11.20	6.78 ±2.55	**5.06 ±2.09**	6.67 ±2.32	5.43 ±1.96	7.56 ±3.86	6.05 ±3.01
Iono	20.71 ±6.05	17.93 ±2.14	19.07 ±4.80	34.21 ±6.40	19.86 ±8.38	15.79 ±6.43	22.00 ±7.10	19.65 ±5.23	17.65 ±6.95	**14.39 ±6.51**
Mass	19.69 ±2.66	26.74 ±1.95	20.76 ±3.94	20.76 ±3.31	20.47 ±3.32	**19.45 ±3.51**	21.12 ±3.00	20.24 ±3.14	21.11 ±2.72	20.83 ±3.28
Pima	26.79 ±4.51	31.93 ±1.68	**25.65 ±2.33**	34.45 ±3.86	26.17 ±3.58	26.04 ±4.00	27.69 ±3.17	32.47 ±3.64	27.73 ±6.83	26.30 ±4.68
SPECT	22.08 ±5.71	18.38 ±4.14	18.02 ±5.21	18.49 ±5.89	22.55 ±13.09	22.74 ±4.48	17.83 ±6.07	24.46 ±6.18	20.51±7.33	**17.17 ±5.43**
Transfusion	24.60 ±3.05	37.38 ±1.56	23.50 ±3.01	24.17 ±2.69	27.87 ±4.29	32.50 ±5.04	23.90 ±2.87	24.49 ±2.53	23.78 ±4.37	**23.28 ±2.72**
WDBC	7.59 ±7.56	17.89 ±2.26	6.49 ±3.88	12.02 ±5.56	4.82 ±2.66	**4.17 ±1.39**	5.96 ±2.47	9.00 ±2.86	5.98 ±3.33	4.96 ±2.43
	Results for 20% of labeled data									
Datasets	AdaBoost	NCL	MeanS3VM	ClusterReg	Manifold	SS-ELM	UDEED	SemiBoost	RegBoost	*SemiNCL*
Australian	16.23 ±3.44	27.17 ±2.03	14.93 ±3.08	21.09 ±8.54	14.24 ±2.65	14.13±2.68	14.42 ±2.62	15.12 ±3.88	17.37 ±5.21	**14.10 ±2.58**
BUPA	35.29 ±5.47	36.89 ±2.23	35.72 ±5.93	40.87 ±6.12	40.58 ±6.73	**35.22 ±5.57**	40.00 ±6.14	39.76 ±4.19	41.60 ±7.58	36.88 ±4.58
German	27.13 ±1.76	35.09 ±1.55	25.58 ±2.45	30.05 ±3.15	30.10 ±4.23	28.35 ±1.73	**24.10 ±2.51**	28.59 ±2.92	26.80 ±5.55	24.76 ±2.40
Haberman	28.93 ±5.61	32.38 ±2.48	24.67 ±5.03	24.75 ±4.96	28.28 ±10.45	33.28 ±10.49	24.59 ±5.16	27.00 ±4.27	27.49 ±7.56	**24.39 ±5.26**
Horse	34.64 ±4.74	33.34 ±2.18	33.45 ±5.71	34.26 ±4.75	33.24 ±5.32	34.12 ±5.86	33.45 ±5.99	**27.03 ±4.36**	33.46 ±10.40	32.80 ±5.22
House	5.23 ±1.65	19.34 ±2.63	5.63 ±2.76	15.17 ±8.86	4.89 ±2.00	4.50 ±1.55	4.83 ±2.19	4.52 ±2.22	4.82 ±3.32	**4.37 ±1.58**
Iono	15.79 ±5.07	17.86 ±2.63	17.29 ±4.77	34.07 ±6.39	17.79 ±7.38	13.93 ±5.26	18.07 ±5.49	13.56 ±4.95	13.10 ±5.92	**10.82±3.85**
Mass	18.88 ±2.72	26.84 ±2.29	20.08 ±3.07	20.52 ±2.77	19.87 ±2.78	**18.83 ±3.19**	20.73 ±2.73	19.68 ±2.19	19.41 ±3.53	19.15 ±2.55
Pima	25.81 ±3.68	31.92 ±1.86	23.96 ±2.20	34.81 ±3.32	**23.54 ±3.50**	24.90 ±2.87	26.40 ±3.51	29.90 ±3.92	26.18 ±7.76	23.91 ±3.15
SPECT	19.06 ±6.21	16.14 ±2.67	18.02 ±5.21	18.68 ±4.82	20.47 ±8.33	24.72 ±6.95	16.79 ±5.42	19.79 ±5.82	18.75 ±3.58	**15.66 ±4.33**
Transfusion	22.40 ±2.54	37.50 ±1.20	23.67 ±2.71	24.17 ±2.69	24.83 ±3.44	33.50 ±4.05	23.37 ±3.06	24.29 ±2.00	25.45 ±4.65	**22.18 ±3.65**
WDBC	5.22 ±2.66	17.72 ±2.25	**4.03 ±2.46**	12.19 ±4.66	4.43 ±2.02	4.14 ±1.62	6.23 ±2.13	9.15 ±2.41	5.68 ±2.92	4.96 ±2.21



(a) (b) (c) (d)

Fig. 4. Result of the Friedman test for comparing the performance of SemiNCL on 10 regression data sets. The dots indicate the average ranks, the bars indicate the critical difference with the Bonferroni-Dunn test at significance level 0.05, and the compared methods having nonoverlapped bars are significantly different. (a) 10%. (b) 30%. (c) 50%. (d) 70%.



(a) (b) (c)

Fig. 5. Result of the Friedman test for comparing the performance of SemiNCL on 12 classification data sets. The dots indicate the average ranks, the bars indicate the critical difference with the Bonferroni-Dunn test at significance level 0.05, and the compared methods having nonoverlapped bars are significantly different. (a) 5%. (b) 10%. (c) 20%.

For classification tasks, the differences of SemiNCL versus UDEED (except when the label rate is 5%), Manifold (only when the label rate is 20%), ClusterReg, and NCL are greater than the critical difference, so the differences are significant. Thus, it means that SemiNCL is significantly better than UDEED (except when label rate is 5%), Manifold (only when

the label rate is 20%), ClusterReg, and NCL in these cases.

The differences between SemiNCL and COREG under the label ratios 0.3 and 0.7, the difference between SemiNCL and CoLSR under the label ratio 0.7, the difference between SemiNCL and UDEED under the label ratio 0.05, and the differences between SemiNCL and Manifold under the label ratios 0.05 and 0.10, are below the critical difference, which means that the difference between SemiNCL and COREG/CoLSR/UDEED/Manifold are not significant under these label rates.

We propose two possible reasons why SemiNCL outperforms other semisupervised algorithms in most cases.

1) The assumption in SemiNCL is weaker than that of other SSL algorithms. In the formulation of SemiNCL, we do not explicitly specify any assumptions. The underlying assumption of SemiNCL would be a kind of "smoothness" in the data space, which encourages the diversity using unlabeled data. The graph-based algorithms use a stronger "smoothness" assumption to generate a data graph, and they need to specify additional parameters, such as the number of nearest neighbors and the RBF kernel parameter, to construct a data graph, in addition to the usual regularization parameter γ_A and manifold regularization parameter γ_I. This would lead to extra efforts to search for a good combination of parameters and sometimes this will result in suboptimal solutions. In contrast, SemiNCL does not need to construct a data graph and only assumes that the data space is relatively smooth.
2) As the ensemble learner, NCL is a stronger regressor, which was pointed out before [9], [34], [35] and validated by this paper (see Table II). After incorporating the unlabeled data with a weak assumption of smooth data space, SemiNCL could achieve a better performance than NCL, especially when few labeled and many unlabeled data are presented.

E. Scalability and Computational Complexity

This section presents a scalability study using SemiNCL and other methods. First, we show the generalization error and the computational time required for relatively large data sets. Then, the computational and memory complexity are analyzed.

For regression problems, we compare the computational time and generalization error of SemiNCL, Manifold [1], COREG [51], and CoRLSR [5] using two relatively large data sets: sarcos and kin40k. For classification problems, we compare the computational time and generalization error of SemiNCL, MCSSB [18], RegBoost [14], and ClusterReg [45]. Table V summarizes such data sets. We randomly selected 100 labeled instances for each data set.

Figs. 6 and 7 present generalization error and computational time[15] for sarcos, kin40k, secStr, and acoustic data sets, respectively. For each step in these plots, new unlabeled instances are randomly chosen and included in the previous training set.

TABLE V
SUMMARY OF LARGE DATA SETS

Type	Datasets	# instances	# attributes
Regression	sarcos	48933	27
Regression	kin40k	40000	8
Classification	SecStr	83679	315
Classification	Acoustic	98528	50

According to Fig. 6, the MSE obtained using a large number of unlabeled samples tends to be lower than those obtained using a small number of unlabeled instances given a fixed number of labeled instances. These results confirm the usefulness of unlabeled instances to improve the regression performance when the unlabeled instances came from the same input distribution as the labeled samples.

The running time of Manifold increases when a large data set presents. Since the expensive computational part of Manifold is the construction of the data graph, it consumes similar time for data with different label rates. Since CoRLSR is a kernel method that utilizes all the data points, it is sensitive to the size of data set.

In Fig. 6, SemiNCL scales well with the increase of the unlabeled data. While the Manifold approach costs the most time and it will run out of memory with more unlabeled data. CoRLSR also suffers from the huge memory consumption problem and it will stop with more unlabeled data coming. Although COREG is able to handle large data sets, it cannot compete with SemiNCL in terms of generalization performance. Based on these results, it is clear that SemiNCL can obtain less generalization error with good scalability for regression problems.

Similarly, in Fig. 7, the algorithms reduce their generalization error with larger amounts of unlabeled data. However, as depicted in Fig. 7, only SemiNCL was able to handle the full data sets. According to Fig. 7, MCSSB, RegBoost, and ClusterReg fail with a few thousands of instances. MCSSB cannot deliver comparable accuracy to other algorithms. MCSSB updates each instance weight with the consideration of all other unlabeled points, which means it uses all samples to assign the pseudo-labels for the unlabeled data, which leads to a quadratic growth of computational time with respect to the number of unlabeled samples. Moreover, MCSSB stores a $S \times S$ [16] similarity matrix. Such facts cause the algorithm to fail due to either memory shortage or time usage. RegBoost requires the computation of the exact nearest neighbors, which involves the operation of an $S \times S$ distance matrix. As indicated by Fig. 7, RegBoost starts to demand lots of memory with only small amounts of data, which leads to a rapid increase in computation time at each step of the graph. Therefore, similar to MCSSB, with a certain large number of instances, RegBoost fails due to infeasible running time and memory consumption.

As shown in Figs. 6 and 7, the time requirement of SemiNCL grows linearly with the number of unlabeled samples.

[15]The computational environment is Linux with Intel 4 core 2.5-GB CPU and 4-GB RAM.

[16]$S = N + V$ is the total number of labeled and unlabeled samples.

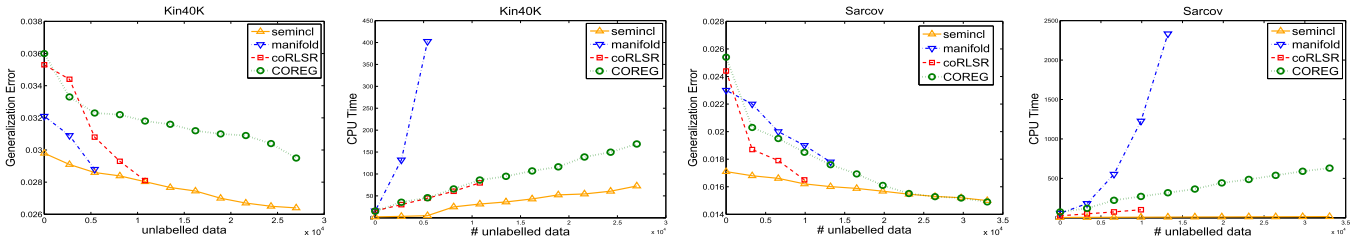


Fig. 6. Error rate and CPU time of four SSL algorithms implementations on kin40K and sarcov data sets with varying unlabeled data and 100 labeled points.

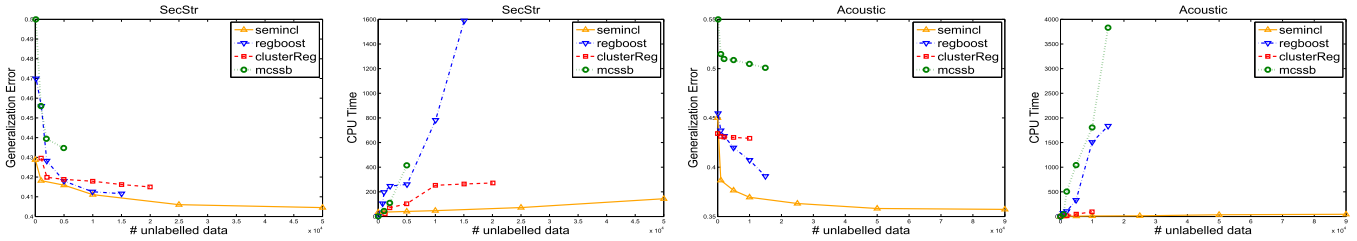


Fig. 7. Error rate and CPU time of four semisupervised ensemble learning algorithms implementations on SecStr and Acoustic data sets with varying unlabeled data and 100 labeled points.

TABLE VI

COMPARISONS OF SEMINCL IN TERMS OF COMPUTATIONAL COMPLEXITY AND MEMORY COMPLEXITY

Algorithm	Computational Complexity	Memory Complexity
MCSSB	$\mathcal{O}(TS^2 + TR^3)$	$\mathcal{O}(S^2)$
RegBoost	$\mathcal{O}(VS \log S + TVS + TR^3)$	$\mathcal{O}(S^2)$
ClusterReg	$\mathcal{O}(VS \log S + TVS)$	$\mathcal{O}(S^2)$
SemiNCL	$O(W(H+M)S + WH^2)$	$O(WS)$

where T is the number of base learners, V is the number of neighbors, R is the number of resampling samples, H is the average number of weights in each RBF network, M is the number of RBF networks, W ($W = M \cdot H$) is the total number of weights in all RBF networks, and S is the total number of labeled and unlabeled samples.

In this sense, SemiNCL is suitable for (relatively) large-scaled data sets, delivering good generalization efficiently without compromising memory usage, which is attributed to the approximation technique in Section III-C that improves the computational efficiency and avoids the drawback of RegBoost and MCSSB with respect to high memory consumption.[17]

Table VI summarizes the computational and memory complexity for SemiNCL, MCSSB, RegBoost, and ClusterReg. From Table VI, we observe that the computational complexity of SemiNCL is related to the values of M, H, and W. According to the experimental settings in Section IV-A, the value of M is fixed to 25, the average number of hidden nodes $H \approx 7$, thus W ($W = M \cdot H$) is a constant in our experiments. Therefore, the values of M, H, and W are constant and independent with S, and they have little effect on the efficiency of SemiNCL especially for (relatively) large data sets. It is easy to observe that the computational and memory complexity of SemiNCL scale linearly with the number of training samples, and SemiNCL is more efficient than other algorithms.

[17]This drawback also has an impact on the execution time due to the memory overload

V. CONCLUSION

In this paper, a semisupervised ensemble learning algorithm, i.e., SemiNCL algorithm, has been proposed. By introducing a negative correlation term for both labeled and unlabeled data, SemiNCL can effectively exploit the unlabeled samples to encourage ensemble diversity on unlabeled data without sacrificing the accuracy on labeled data. The traditional NCL is optimized by conjugate gradient, which requires additional computational time. In this paper, an accelerated SemiNCL is derived from the distributed least square algorithm, which provides a closed-form solution and avoids expensive computational and memory costs, making SemiNCL scalable for relatively large-scaled data sets. We also provide the theoretical analysis of the negative correlation imposing parameters, and derive a bound for the parameters based on the analysis of Hessian matrices. State-of-the-art semisupervised and supervised algorithms have been used to compare with SemiNCL, and the experimental results on regression and classification tasks demonstrate that SemiNCL performs very well and outperforms others in general. We apply SemiNCL to several relatively large data sets and analyze the computational and memory complexity, and the results demonstrate the effectiveness of SemiNCL on large data sets.

We discuss the underlying assumption of SemiNCL, i.e., the unlabeled data should follow a similar distribution or tendency with the labeled data, which is a kind of weak smoothness assumption in the data space. By incorporating this weak assumption, SemiNCL promotes the ensemble diversity and achieves better performance in comparison with other SSL algorithms across a number of different data sets, especially when few labeled and many unlabeled data are presented.

REFERENCES

[1] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2006.

[2] K. P. Bennett, A. Demiriz, and R. Maclin, "Exploiting unlabeled data in ensemble methods," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2002, pp. 289–296.

[3] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory (COLT)*, 1998, pp. 92–100.

[4] U. Brefeld, C. Büscher, and T. Scheffer, "Multi-view discriminative sequential learning," in *Proc. 16th Eur. Conf. Mach. Learn. (ECML)*, 2005, pp. 60–71.

[5] U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel, "Efficient co-regularised least squares regression," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 137–144.

[6] U. Brefeld and T. Scheffer, "Semi-supervised learning for structured output variables," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 145–152.

[7] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[8] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: A survey and categorisation," *Inf. Fusion*, vol. 6, no. 1, pp. 5–20, 2005.

[9] G. Brown, J. L. Wyatt, and P. Tiňo, "Managing diversity in regression ensembles," *J. Mach. Learn. Res.*, vol. 6, pp. 1621–1650, Dec. 2005.

[10] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proc. 10th Int. Workshop Artif. Intell. Statist. (AISTATS)*, 2005, pp. 57–64.

[11] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.

[12] H. Chen and X. Yao, "Regularized negative correlation learning for neural network ensembles," *IEEE Trans. Neural Netw.*, vol. 20, no. 12, pp. 1962–1979, Dec. 2009.

[13] H. Chen and X. Yao, "Multiobjective neural network ensembles based on regularized negative correlation learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 12, pp. 1738–1751, Dec. 2010.

[14] K. Chen and S. Wang, "Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 129–143, Jan. 2011.

[15] L. Chen, I. W. Tsang, and D. Xu, "Laplacian embedded regression for scalable manifold regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 6, pp. 902–915, Jun. 2012.

[16] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Netw.*, vol. 2, no. 2, pp. 302–309, Mar. 1991.

[17] R. Collobert and S. Bengio, "SVMTorch: Support vector machines for large-scale regression problems," *J. Mach. Learn. Res.*, vol. 1, pp. 143–160, Sep. 2001.

[18] F. d'Alché-Buc, Y. Grandvalet, and C. Ambroise, "Semi-supervised MarginBoost," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2002, pp. 553–560.

[19] H. H. Dam, H. A. Abbass, C. Lokan, and X. Yao, "Neural-based learning classifier systems," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 1, pp. 26–39, Jan. 2008.

[20] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

[21] O. J. Dunn, "Multiple comparisons among means," *J. Amer. Statist. Assoc.*, vol. 56, no. 293, pp. 52–64, 1961.

[22] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[23] Y. Freund, R. E. Schapire, and N. Abe, "A short introduction to boosting," *J. Japn. Soc. Artif. Intell.*, vol. 14, no. 5, pp. 771–780, 1999.

[24] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Statist. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.

[25] A. Fujino, N. Ueda, and K. Saito, "Semisupervised learning for a hybrid generative/discriminative classifier based on the maximum entropy principle," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 424–437, Mar. 2008.

[26] J. B. Gomm and D. L. Yu, "Selecting radial basis function network centers with recursive orthogonal least squares training," *IEEE Trans. Neural Netw.*, vol. 11, no. 2, pp. 306–314, Mar. 2000.

[27] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, "Deformed graph Laplacian for semisupervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2261–2274, Oct. 2015.

[28] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014.

[29] M. M. Islam, X. Yao, and K. Murase, "A constructive algorithm for training cooperative neural network ensembles," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 820–834, Jul. 2003.

[30] B. Jiang, H. Chen, B. Yuan, and X. Yao, "Scalable graph-based semi-supervised learning through sparse Bayesian model," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2758–2771, Dec. 2017.

[31] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. 16th Int. Conf. Mach. Learn. (ICML)*, 1999, pp. 200–209.

[32] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 3581–3589.

[33] Y. Li, J. T. Kwok, and Z.-H. Zhou, "Semi-supervised learning using label mean," in *Proc. 26nd Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 633–640.

[34] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Netw.*, vol. 12, no. 10, pp. 1399–1404, Dec. 1999.

[35] Y. Liu and X. Yao, "Simultaneous training of negatively correlated neural networks in an ensemble," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 6, pp. 716–725, Dec. 1999.

[36] Y. Liu, X. Yao, and T. Higuchi, "Evolutionary ensembles with negative correlation learning," *IEEE Trans. Evol. Comput.*, vol. 4, no. 4, pp. 380–387, Nov. 2000.

[37] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu, "SemiBoost: Boosting for semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 2000–2014, Nov. 2009.

[38] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Netw.*, vol. 6, no. 4, pp. 525–533, Nov. 1993.

[39] B. Nadler, N. Srebro, and X. Zhou, "Semi-supervised learning with the graph Laplacian: The limit of infinite unlabelled data," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2009, pp. 1330–1338.

[40] M. Lichman, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 2003. [Online]. Available: http://archive.ics.uci.edu/ml

[41] E. Riloff, J. Wiebe, and T. Wilson, "Learning subjective nouns using extraction pattern bootstrapping," in *Proc. 7th Conf. Natural Lang. Learn. (CONNL)*, 2003, pp. 25–32.

[42] R. G. F. Soares, H. Chen, and X. Yao, "A cluster-based semisupervised ensemble for multiclass classification," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 1, no. 6, pp. 408–420, Dec. 2017.

[43] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *Proc. 7th IEEE Workshops Appl. Comput. Vis. (WACV)*, Jan. 2005, pp. 29–36.

[44] B. M. Shahshahani and D. A. Landgrebe, "The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 5, pp. 1087–1095, Sep. 1994.

[45] R. G. F. Soares, H. Chen, and X. Yao, "Semisupervised classification with cluster regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 11, pp. 1779–1792, Nov. 2012.

[46] E. K. Tang, P. N. Suganthan, and X. Yao, "An analysis of diversity measures," *Mach. Learn.*, vol. 65, no. 1, pp. 247–271, Oct. 2006.

[47] X. Yao, M. Fischer, and G. Brown, "Neural network ensembles and their application to traffic flow prediction in telecommunications networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2001, pp. 693–698.

[48] S. Zhai, T. Xia, Z. Li, and S. Wang, "A direct boosting approach for semi-supervised classification," in *Proc. 24th Int. Conf. Artif. Intell. (IJCAI)*, 2015, pp. 4025–4032.

[49] K. Zhang, L. Lan, J. Kwok, S. Vucetic, and B. Parvin, "Scaling up graph-based semisupervised learning via prototype vector machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 444–457, Mar. 2015.

[50] M.-L. Zhang and Z.-H. Zhou, "Exploiting unlabeled data to enhance ensemble diversity," *Data Mining Knowl. Discovery*, vol. 26, no. 1, pp. 98–129, Jan. 2013.

[51] Z. H. Zhou and M. Li, "Semisupervised regression with cotraining-style algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 11, pp. 1479–1493, Nov. 2007.

[52] X. Zhu, "Semi-supervised learning literature survey," Dept. Comput. Sci., Univ. Wisconsin, Madison, WI, USA, Tech. Rep. TR 1530, 2007. [Online]. Available: http://www.cs.wisc.edu/ jerryzhu/pub/ssl_survey.pdf

[53] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 912–919.

Huanhuan Chen (M'09–SM'16) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2004, and the Ph.D. degree in computer science from the University of Birmingham, Birmingham, U.K., in 2008.

He is currently a Full Professor with the School of Computer Science and Technology, USTC. His current research interests include neural networks, Bayesian inference, and evolutionary computation.

Dr. Chen was a recipient of the 2015 International Neural Network Society Young Investigator Award, the 2012 IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation Award, the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award (bestowed in 2011 and only one paper in 2009), and the 2009 British Computer Society Distinguished Dissertations Award. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE.

Bingbing Jiang received the B.Sc. degree from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2014. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China.

His current research interests include Bayesian learning, semisupervised learning, and feature selection.

Xin Yao (F'03) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1982, the M.Sc. degree from the North China Institute of Computing Technology, Beijing, China, in 1985, and the Ph.D. degree from USTC, in 1990.

He is currently a Chair Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China, and a part-time Professor of computer science at the University of Birmingham, Birmingham, U.K. His current research interests include evolutionary computation and ensemble learning.

Dr. Yao was a recipient of the 2001 IEEE Donald G. Fink Prize Paper Award, the 2010 and 2016 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Awards, the 2011 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award, and several other Best Paper Awards. He was the Editor-in-Chief (2003–2008) of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the President (2014–2015) of the IEEE Computational Intelligence Society (CIS). He was a Distinguished Lecturer in the IEEE CIS.