Efficient Cluster-Based Boosting for Semisupervised Classification

Rodrigo G. F. Soares, Huanhuan Chen, *Senior Member, IEEE*, and Xin Yao, *Fellow, IEEE*

***Abstract*— Semisupervised classification (SSC) consists of using both labeled and unlabeled data to classify unseen instances. Due to the large number of unlabeled data typically available, SSC algorithms must be able to handle large-scale data sets. Recently, various ensemble algorithms have been introduced with improved generalization performance when compared to single classifiers. However, existing ensemble methods are not able to handle typical large-scale data sets. We propose efficient cluster-based boosting (ECB), a multiclass SSC algorithm with cluster-based regularization that avoids generating decision boundaries in high-density regions. A semisupervised selection procedure reduces time and space complexities by selecting only the most informative unlabeled instances for the training of each base learner. We provide evidences to demonstrate that ECB is able to achieve good performance with small amounts of selected data and a relatively small number of base learners. Our experiments confirmed that ECB scales to large data sets while delivering comparable generalization to state-of-the-art methods.**

***Index Terms*— Cluster-based regularization, ensemble learning, multiclass classification, semisupervised classification.**

I. Introduction

LABELING instances may require the allocation of expensive resources, e.g., human expertise and time. On the other hand, obtaining large amounts of unlabeled points can be cheap and straightforward. Semisupervised classification (SSC) algorithms learn from both labeled and unlabeled instances.

In order to use the unlabeled data distribution, SSC algorithms employ various assumptions [1]. The smoothness assumption states that if a pair of points is similar, they are likely to yield similar outputs. The cluster assumption assumes that classes are often separated by a low-density region. If two instances belong to one cluster, it is probable that they share class probabilities. And the manifold assumption states that the true structure of the data lies in a low-dimensional manifold embedded in the high-dimensional data space and, by using such manifold instead of the original structure, the classifier would have better generalization accuracy.

The typical large number of unlabeled instances has a great impact on the computational time of existing semisupervised classifiers. Among methods that implement the cluster assumption (cluster-based algorithms), transductive support vector machines (TSVMs) [2] is a popular choice. However, it requires time $\mathcal{O}(m^3)$ where m is the number of instances. Classifiers based on the manifold assumption (manifold-based algorithms) are also time consuming, they require $\mathcal{O}(m^3)$ or $\mathcal{O}(vm^2)$ [3], [4]. For such binary classifiers, this issue is aggravated in multiclass SSC by the additional computational time required by decomposition approaches, such as one-versus-all.

Ensemble learning has been successfully employed in both supervised [5] and semisupervised [6], [7] classification to improve generalization when compared to single classifiers. However, the use of existing ensemble techniques in large SSC data sets is limited due to time and memory requirements. For example, RegBoost [7] is a binary ensemble classifier that, if implemented with support vector machines (SVMs),[1] requires $\mathcal{O}(vm \log m + \tau s^3 + \tau vu)$, where τ is the number of base learners (boosting iterations), s is the sample size, and u is the number of unlabeled points. Due to the computation of nearest neighbors, RegBoost requires memory of $\mathcal{O}(m^2)$, which also might prevent its application to large data sets.

The computation complexity of binary ensembles will increase for multiclass classification. A few multiclass ensemble approaches have been proposed [6], [8]. However, despite having implemented the cluster assumption, these algorithms do not take full advantage of cluster assignments [9].

The work of [10] introduces cluster-based boosting (CBoost), a multiclass ensemble approach that uses clustering methods in its regularization technique. Such a classifier is able to overcome possible errors in pseudolabels assignments from the ensemble procedure and is robust to overlapping classes and to the relative situation of few labeled points in a given cluster when the cluster assumption holds. However, the complexity of CBoost is $\mathcal{O}(ec^2h^2u + \tau cvu)$, where c is the number of classes, h is the number of hidden nodes in radial basis function network (RBFN), e is the number of epochs in RBFN, and u is the number of

Manuscript received March 28, 2017; revised November 16, 2017; accepted February 10, 2018. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1000905 and in part by the National Natural Science Foundation of China under Grant 91546116 and Grant 91746209. *(Corresponding author: Huanhuan Chen.)*

R. G. F. Soares is with the Department of Informatics, Federal Rural University of Pernambuco, Recife 52171-900, Brazil (e-mail: rgfsoares@gmail.com).

H. Chen is with UBRI, School of Computer Science, University of Science and Technology of China, Hefei 230027, China (e-mail: hchen@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 (e-mail: x.yao@cs.bham.ac.uk).

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.2018.2809623

[1] SVM is the base classifier recommended by Chen and Wang [7].

unlabeled instances. CBoost may not be applicable to large-scale data sets.

We propose the efficient cluster-based boosting (ECB). ECB is a greedy boosting approach that improves the scalability of ensemble learning for large-scale SSC data sets, while maintaining the generalization ability of ensemble-based methods. The proposed method implements the cluster assumption by using a regularization technique based on cluster posteriors. We tackle large data sets by selecting only the most informative unlabeled points, along with pseudolabels,[2] to form the training set for each base learner. Our experiments demonstrate that ECB delivers good predictive accuracy in large data sets. This paper is an extension of [11], where a comprehensive study on SSC can be found.

A. Contributions

The proposed approach has the following contributions.

1) ECB tackles large-scale data sets by selecting few informative unlabeled instances that will help to produce significant improvement in the decision boundary.
2) ECB employs efficient clustering algorithm and approximates nearest neighbors to reduce time and memory requirements.
3) ECB is robust to overlapping classes and to the position of the few labeled instances in a given cluster when the cluster assumption holds [9].
4) ECB is designed for multiclass problems, so that it does not depend on decomposition techniques.
5) Both ensemble and base classifiers have the same semi-supervised loss function. The base classifiers will also consider the neighborhood of an unlabeled instance when learning its pseudolabel, so that new base learners may be able to overcome possible errors in pseudolabels.

The remainder of this paper is organized as follows. Section II analyses existing methods. Section III introduces ECB, and Section IV shows our experimental studies and discusses our contributions. Section V presents our conclusion.

II. Background

In this section, we discuss the importance of proposing a cluster-based ensemble for multiclass SSC that is able to handle large amounts of data.

The computational complexity of various popular SSC methods prevents their application to large data sets [12]. Manifold-based algorithms require large computational effort due to the construction of graphs to represent the data. Such graphs have labeled and unlabeled points as vertices and labels are assigned to unlabeled vertices based on their neighbors. Zhu and Ghahramani [4] introduced label propagation, where labeled instances are used to assign labels to unlabeled instances in its neighborhood according to a graph. In [13], a transductive version of the k-NN classifier was trained via graphs. In [3], random walks were employed in graphs to assign labels to unlabeled data. The computational complexity of such methods is $\mathcal{O}(m^3)$ or $\mathcal{O}(vm^2)$ [12]. Moreover, such manifold-based algorithms depend on the graph construction, which often is a suboptimal procedure [14]. Typically, these methods are not able to tackle unseen (test) data as they are inherently transductive. Such a limitation can prevent the application of graph-based methods in inductive problems.

Most existing cluster-based algorithms are also computationally intensive. TSVM [2] attempts to find the largest margin between classes by searching for different label assignments for unlabeled data and calculating margins between dense regions of similarly labeled instances. Such a procedure is expensive and requires time of $\mathcal{O}(m^3)$. Later, [15] developed a more efficient implementation of TSVM. If dense regions are overlapping, this classifier might not generate a correct decision boundary in the gap between these regions (clusters). And, in this case, TSVM might be sensitive to the scarce labeled instances in the dense regions [9].

The aforementioned algorithms are binary classifiers. In order to perform multiclass classification, such methods depend on decomposition techniques, e.g., one-versus-one and one-versus-all. Thus, applying these costly algorithms to multiclass classification requires multiple and expensive runs. Such a drawback has a great impact on large-scale data sets [8].

ClusterReg [9] is a multiclass cluster-based single classifier. Such a method uses posterior cluster probabilities in its regularization mechanism. When the cluster assumption holds, this algorithm is capable of delivering good performance in the presence of overlapping classes and it is robust to the position of labeled instances within a cluster. If implemented with RBFN, the complexity of ClusterReg is $\mathcal{O}(ec^2m^2u)$.

Ensemble algorithms, in particular boosting techniques, were successfully employed in SSC [6]–[8]. RegBoost [7] uses three semisupervised learning (SSL) assumptions. In order to implement the cluster assumption, RegBoost uses a kernel density estimation that will penalize the classifier if it does not assign the same class to a pair of similar instances in a high-density region. However, if overlapping high-density regions are present, RegBoost might not find a potentially correct decision boundary between these regions. RegBoost requires time of $\mathcal{O}(vcm \log m + c\tau s^3 + \tau cvu)$ for multiclass classification and, due to search for nearest neighbors, demands memory of $\mathcal{O}(m^2)$, which might be prohibitive for large data sets. Yu *et al.* [16] investigated the use of ensembles in high-dimensional SSC. Such algorithm splits features into several subspaces, builds a graph for each subspace, trains a linear algorithm (base classifier) on each graph and combines these classifiers as an ensemble. The computational complexity of such method is $\mathcal{O}(m^2d + nds + ds^3)$, where d is the original dimensionality and s is the subspace dimensionality.

The methods in [7] and [16] are binary classification algorithms and depend on the reduction of multiclass classification in multiple two-class problems. Such an issue is exacerbated in the training of several base classifiers.

In order to perform ensemble learning in multiclass SSC, [6], [8], and [17] proposed multiclass boosting techniques. Valizadegan *et al.* [6] introduced multiclass

[2]In this paper, pseudolabels are posterior class probabilities that are systematically assigned to unlabeled instances by some classifiers. Pseudolabels might be different from true labels.

semisupervised boosting (MCSSB) algorithm. MCSSB is a multiclass version of the SemiBoost algorithm proposed in [18]. Such an algorithm combines the similarity information among the instances with the classifier predictions in order to generate more reliable pseudolabels. It is a graph-based approach, its objective function possesses three SSL assumptions and it uses supervised base classifiers. The computational complexity of MCSSB is $\mathcal{O}(\tau cu^2 + \tau s^3)$, where s is the number of sampled instances. MCSSB stores a similarity matrix that requires memory of $\mathcal{O}(m^2)$. Such requirements might limit its application to large data sets.

Applying the state-of-the-art algorithms described here to large-scale data sets might be a challenging task due to high computational complexity. Delalleau *et al.* [19] proposed a sampling technique to reduce computational complexity from $\mathcal{O}(m^3)$ to $\mathcal{O}(s^2 m)$, where s is the number of sampled instances. However, such a technique is designed for transductive graph-based algorithms and the experimental results in [1] show that the difference between such an algorithm and uniform random sampling is marginal. Other techniques for increasing efficiency can reduce the time complexity to $\mathcal{O}(s^3)$, where $s < m$, but may reduce generalization [12], [20].

The training of several classifiers limits the use of existing ensemble algorithms in large-scale SSC. This shortcoming is aggravated in the multiclass context [8]. Delalleau *et al.* [19] proposed a sampling procedure to tackle large-scale data sets. However, such a technique is designed for transductive graph-based algorithms. And in the experimental analysis of [1], such a sampling technique did not show considerable improvement over uniform random sampling. Moreover, many SSC methods (ensemble and single classifiers) depend on the calculation of a pairwise distance matrix [6], [9] that requires memory in the order of $\mathcal{O}(m^2)$, which also restrains the application of ensemble methods to large data sets.

In order to address such limitations, we propose an efficient cluster-based multiclass boosting algorithm that maintain comparable generalization with state-of-the-art methods. We instantiated the gradient boosting framework [21] and introduced a basis selection procedure to obtain an efficient classifier [22]. Gradient boosting produces highly robust ensemble classifiers and its instantiation is straightforward, both boosting and selection procedures are based on the steepest descent method [21].

In order to handle large amounts of data, in each iteration, each base classifier is trained with only the most relevant unlabeled instances along with all labeled points.[3] Since ECB depends on the output of a clustering algorithm, we employed the landmark-based spectral clustering (LSC) algorithm to efficiently compute posterior cluster probabilities [23]. We also used the approximation technique introduced in [24] to efficiently obtain nearest neighbors and avoid the expensive computation of pairwise distance matrix. We selected RBFN as base learner due to its effectiveness and efficiency [25].

[3]In SSC, the number of labeled instances may be order of magnitudes smaller than the number of unlabeled points. Therefore, we can avoid sampling and safely use all available labeled data in the training set of base learners.

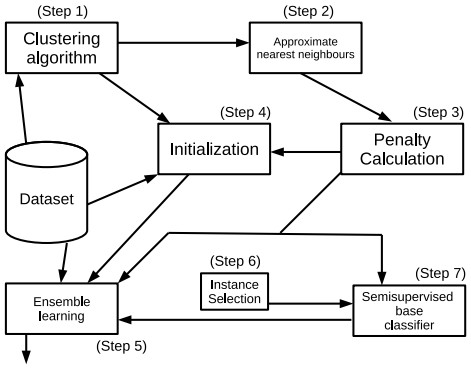


Fig. 1. ECB's architecture.

III. Efficient Cluster-Based Boosting

The training set $X = L \cup U$ is formed of l labeled instances $L = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{l}$ and u unlabeled instances $U = \{\mathbf{x}_n\}_{n=l+1}^{m}$, often $u \gg l$, and $m = l + u$. Labels $\mathbf{y}_n$ are class probabilities, $0 \leq y_{ni} \leq 1$ and $\sum_{i=1}^{c} y_{ni} = 1$. SSC aims to improve the generalization of a classifier in comparison to using only the labeled data L.

In this section, we introduce the ECB algorithm. Fig. 1 shows the general architecture of the proposed method. Its steps are as follows.

1) Extract posterior cluster probabilities from an efficient clustering algorithm.
2) An approximation technique is employed to find the nearest neighbors for all unlabeled instances.
3) The initialization procedure assigns initial label estimates to unlabeled instances.
4) ECB selects a subset of the unlabeled instances and their label estimates, along with all labeled instances, to compose the initial training set in order to improve the efficiency of the ensemble training.
5) The training of the first base classifier is performed on the initial training set.
6) The ensemble algorithm trains a number of semisupervised base classifiers with unlabeled instances that are selected and greedily labeled according to gradient descent at each iteration.
7) The ensemble weights and combines the outputs of all trained base classifiers to perform predictions for unseen instances.

A. Multiclass Loss Function With Cluster Regularization

We employ a cluster-based loss function that consists of two terms: supervised cost and cluster regularization [9]. Since we are focusing on multiclass SSC, we use cross-entropy and softmax functions to compose the loss function [26].

We assume that a clustering algorithm produces a partition $\mathbf{Q} = [q_{ni}]_{m \times g}$ with g clusters and m instances. The vector $\mathbf{q}_n$ is the nth row of $\mathbf{Q}$, which has the posterior cluster probabilities of instance n. This vector sums to one and n is associated with the group with highest probability.

Equation (1) defines the multiclass loss function with cross entropy

$$\mathcal{L}(\mathbf{F}, \mathbf{Y}) = -\sum_{n=1}^{m}\sum_{i=1}^{c}\left\{\frac{I_{nL}}{l} y_{ni} \ln(f_{ni}) + \frac{I_{nU}\lambda \max(\mathbf{q}_n)}{u}\hat{y}_{ni}\ln(f_{ni})\right\} \quad (1)$$

where $I_{nL} = 1$ if $n \in L$ and 0 otherwise; $I_{nU} = 1$ if $n \in U$ and 0 otherwise. We denote the posterior class probabilities, for class i and instance n, produced by single classifiers and ensembles as $\mathbf{F} = [f_{ni}]_{m\times c}$ and $\mathbf{F}' = [f'_{ni}]_{m\times c}$, respectively.[4] The matrix $\mathbf{Y} = [y_{ni}]_{l\times c}$ contains the true class memberships of labeled instances. The parameter λ controls the tradeoff between the supervised loss and semisupervised regularization, c is the number of classes and $\max(\mathbf{q}_n)$ returns the maximum cluster likelihood in $\mathbf{q}_n$.

For unlabeled instances, we assign an estimate label to each unlabeled point according to its penalty values and neighborhood (derived from cluster memberships). Equation (2) denotes the estimated class for an unlabeled instance

$$\hat{y}_{ni} = \frac{\sum_{j\in V_n} p(\mathbf{q}_j, \mathbf{q}_n)\gamma_{ji}}{\sum_{j\in V_n} p(\mathbf{q}_j, \mathbf{q}_n)} \quad (2)$$

where $\hat{y}_{ni}$ is the estimated probability of class i for instance n. The set V_n has the nearest neighbors of n. The penalty $p(\mathbf{q}_j, \mathbf{q}_n)$ is calculated according to cluster memberships. Higher similarity between instaces n and j incurs a higher penalty for that pair. The value γ_{ji} may be either the true label y_{ji} if j is labeled or the ensemble prediction f'_{ji} if j is unlabeled. When j is unlabeled, γ_{ji} is an estimate of class i for instance j. The pseudolabel $\hat{y}_{ni}$ is the weighted average of current pseudolabels present in the neighborhood of n. In contrast to [9], we opted to normalize $\hat{y}_{ni}$ by the sum of penalties assigned to each neighbor j, instead of using only the number of neighbors as a scaling factor. This approach weights the influence of neighboring instances on putative labels by their similarity. In this sense, unlabeled instances will have label estimates $\hat{y}_{ni}$ more similar to each γ_{ji} (either label estimates or true labels) of its closest instances.

Equation (3) computes the penalty between n and j. It maps similarity into penalization following a Gaussian curve:

$$p(\mathbf{q}_j, \mathbf{q}_n) = \exp\left(\frac{-||\mathbf{q}_n - \mathbf{q}_j)||^2}{\sigma^2}\right). \quad (3)$$

The width σ regulates the steepness of such a mapping. With lower σ, only most similar instances will have high penalty. It controls the extent in which the decision boundary avoids clusters. In this sense, contrasting predictions for instances with similar cluster memberships will be more severely regularized.

B. Approximate Nearest Neighbors and Large-Scale Clustering

In the proposed method, the clustering algorithm has a great impact on both generalization ability and efficiency. Our preliminary experiments showed that LSC [23] can handle large data sets and leads to good generalization accuracy when compared to other clustering algorithms, e.g., k-means and the spectral method in [27]. Its time complexity is $\mathcal{O}(tpmd + p^3 + p^3 d)$, where t is the number of iterations in k-means and p is the number of landmarks and $p \ll m$. Thus, we selected LSC as the clustering algorithm employed to produce matrix $\mathbf{Q}$.

Semisupervised methods often seek the labels in the neighborhood of an instance in order to assign pseudolabels. The construction of such a neighborhood requires the calculation of all pairwise distances in a $n \times n$ matrix and the search for all neighbors, which requires time of $\mathcal{O}(\upsilon m \log m)$ and memory of $\mathcal{O}(m^2)$ [28], where υ is the number of neighbors.

Since there is a number of approximation techniques that can be employed to reduce computational complexity, we chose a method that automatically selects a suitable approximate algorithm to find nearest neighbors for each instance (row) represented in $\mathbf{Q}$ with less time requirement [24].[5] With the use of such an approximation technique, ECB reduces the memory requirement to $\mathcal{O}(\upsilon m)$, and $\upsilon \ll m$.

The output of such an algorithm is a matrix with the distances from each instance in $\mathbf{Q}$ to its υ neighbors. We use such a matrix to calculate the penalty values. The soft partition arising from the clustering algorithm is used to generate regularization and, therefore, to implement the cluster assumption in our algorithm. ECB employs the smoothness assumption by penalizing the classifier if it assigns different labels to similar points, as denoted by $-\hat{y}_{ni}\ln(f_{ni})$.

C. Initialization Procedure

For many SSC algorithms, if the classifier assigns the same class to every unlabeled instance, the training error will be in a useless local optimum [12]. This is due to the loss function comparing a predicted output with similar output of its neighbors. In order to overcome this local optimum, we use an initialization procedure that employs the distribution of labeled points in a cluster to assign initial label estimates to unlabeled data [9].

We use the sum of labels present in a cluster, weighted by penalty values $p(\mathbf{q}_j, \mathbf{q}_n)$, to assign pseudolabels to unlabeled instances in such cluster [7]. If there are no labeled points in a cluster, equal probabilities will be assigned to each class. For class i of unlabeled instance n in each cluster C we have

$$\hat{y}_{ni} = \frac{\sum_{j\in C} I_{jL} * p(\mathbf{q}_j, \mathbf{q}_n) * y_{ji}}{\sum_{j\in C} I_{jL} * p(\mathbf{q}_j, \mathbf{q}_n)}. \quad (4)$$

The initialization procedure consists of training an initial base classifier for a small number of iterations[6] with the pseudolabels $\hat{y}_{ni}$.

[4]Ensemble-related values are distinguished by the prime symbol.

[5]From the available distances in [24], we selected euclidean distance for searching nearest neighbors. Other distances may be used to improve performance.

[6]We fixed the number of iterations of pretraining at 10. According to our preliminary experiments, different numbers of iterations did not improve the generalization ability of our classifier.

D. RBFN as Base Learner

We chose RBFN as base classifier as it can be efficient [25] and easily adapted to our method. The unsupervised phase of RBFN training consists of the center selection and width estimation described in Section III-F. Its semisupervised training phase has the loss function in (1), with the addition of a weight regularization term. The output (activation) function is $f_{ni} = \text{softmax}(z_{ni})$, where $\mathbf{Z} = [z_{ni}]_{m\times c}$ represents the net input of the output nodes, as in (8). Equation (5) presents the loss function for RBFN

$$\mathcal{L}(\mathbf{F},\mathbf{Y}) = -\sum_{n=1}^{m}\sum_{i=1}^{c}\left\{\frac{I_{nL}}{l}y_{ni}\ln(f_{ni}) + \frac{I_{nU}\lambda\max(\mathbf{q}_n)}{u}\hat{y}_{ni}\ln(f_{ni}) - \alpha\frac{\mathbf{w}_i^T\mathbf{w}_i}{2}\right\} \tag{5}$$

where $\mathbf{w}_i$ is the weight vector for output node (class) i and α controls the amount of weight regularization.

The semisupervised training algorithm for our base learner is the iteratively reweighted least-square (IRLS) method [26]. IRLS consists of e Newton–Raphson steps (epochs) to update network weights, in the following equation:

$$\Delta\mathbf{w}_j = -\mathbf{H}^{-1} * \nabla_{\mathbf{w}_j}\mathcal{L}(\mathbf{F},\mathbf{Y}) \tag{6}$$

where $\mathbf{H}$ is the Hessian matrix and $\nabla_{\mathbf{w}_j}\mathcal{L}$ is the gradient of loss function $\mathcal{L}$ with respect to (w.r.t.) weight vector $\mathbf{w}_j$.

Equation (7) shows the gradient of $\mathcal{L}$ for point n. The vector $\mathbf{f}_n$ is the nth row in $\mathbf{F}$

$$\nabla_{\mathbf{w}_j}\mathcal{L}(\mathbf{f}_n,\mathbf{y}_n) = \frac{I_{nL}}{l}(f_{nj}-y_{nj})\boldsymbol{\phi}_n + \frac{I_{nU}\lambda\max(\mathbf{q}_n)}{u}(f_{nj}-\hat{y}_{nj})\boldsymbol{\phi}_n + \alpha\mathbf{w}_j \tag{7}$$

where $\boldsymbol{\phi}_n$ is the output column vector of hidden nodes.

The Hessian is a block matrix $\mathbf{H} = [H_{jk}]_{hc\times hc}$ (h is the number of hidden nodes), where each block is

$$H_{jk} = \left[\frac{\partial^2\mathcal{L}}{\partial\mathbf{w}_j\partial\mathbf{w}_k}\right] = \sum_{n=1}^{m}\left\{\left(\frac{I_{nL}}{l}+\frac{I_{nU}\lambda\max(\mathbf{q}_n)}{u}\right) * f_{nj}(\delta_{jk}-f_{nk}) * \boldsymbol{\phi}_n\boldsymbol{\phi}_n^T + \alpha\right\}.$$

The update rule in (6) is iterated until a stopping criterion (e.g., increase in validation error) is met. The width of a hidden node in an RBFN is estimated by the median of the pairwise Euclidean distances among instances in the cluster to which that center belongs.

E. Boosting for Large-Scale Multiclass Classification

Following [21] and [26], we employ (8) to transform linear outputs $\mathbf{Z}' = [z'_{ni}]_{m\times c}$ from an ensemble into posterior class probabilities $\mathbf{F}'$

$$f'_{ni} = \text{softmax}(z'_{ni}) = \frac{\exp(z'_{ni})}{\sum_j^C \exp(z'_{nj})}. \tag{8}$$

Unlike original gradient boosting, a base classifier, trained with pseudolabels $\hat{\boldsymbol{y}}_n$ delivered by the initialization procedure, is assigned to initial ensemble $\mathbf{Z}'(0)$, where we denote linear ensemble outputs as an iterative function. As indicated in our preliminary experiments, this initialization delivered better results than simply assigning a constant, e.g., $\mathbf{Z}'(0) = 0$.

We calculate the derivative of $\mathcal{L}(\mathbf{F}'(t),\mathbf{Y})$ w.r.t. $z'_{ni}(t)$ to obtain the current residuals r_{nj} for class i that will be used to train a new base classifier $\mathbf{F}$. Such residuals are computed as in the following equation:

$$r_{ni} = -\frac{I_{nL}}{l}(f'_{ni}(t)-y_{ni}) - \frac{I_{nU}\lambda\max(\mathbf{q}_n)}{u}(f'_{ni}(t)-\hat{y}_{ni}). \tag{9}$$

We select a subset S of size s from U and include all labeled instances, according to the procedure in Section III-F in order to approximate the information in X as accurately and efficiently as possible. A new base learner is fit to the residuals r_{nj} in probability scale, i.e., $\hat{y}_{nj} = \text{softmax}(r_{nj})$, where $\mathbf{x}_n \in S$. The residuals from S provide an approximation of the direction of the gradient $\nabla\mathcal{L}$. Each new base classifier represents a greedy step to approximately minimize $\mathcal{L}$.

For each class, we perform a line search with a single Newton–Raphson step in order to find an appropriate fit for the new base learner in the current ensemble. This line search consists of optimizing the base learner weight $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_c)$, which is initially 0, as shown in the following equation:

$$\boldsymbol{\beta} = -\mathbf{H}^{-1} * \nabla_{\boldsymbol{\beta}}\mathcal{L}(\mathbf{F}'(t),\mathbf{Y}). \tag{10}$$

The gradient $\nabla_{\beta_i}\mathcal{L}$ for each class i is[7]

$$\nabla_{\beta_i}\mathcal{L}(\mathbf{F}'(t),\mathbf{Y}) = \sum_{n=1}^{m}\left\{\frac{I_{nL}}{l}\left(f'_{ni}(t)-y_{ni}\right)z_{ni} + \frac{I_{nU}\lambda\max(\mathbf{q}_n)}{u}\left(f'_{ni}(t)-\hat{y}_{ni}\right)z_{ni}\right\}$$

and Hessian matrix $\mathbf{H}$ is

$$H_{jk} = \sum_{n=1}^{m}\left(\frac{I_{nL}}{l}+\frac{I_{nU}\lambda\max(\mathbf{q}_n)}{u}\right) * f'_{nj}(t)(\delta_{jk}-f'_{nk}(t)) * z_{nj}z_{nk}$$

where $\delta_{jk} = 1$ if $j = k$ and 0 otherwise.

The base classifier is included in the current ensemble following the rule in (11), where η is a learning rate. Such a learning rate might reduce overfitting by decreasing the influence of newly trained base learners on the ensemble

$$z'_{ni}(t+1) = z'_{ni}(t) + \eta\beta_i z_{ni}. \tag{11}$$

Several greedy steps of gradient descent are performed until a stopping criterion is met, e.g., a fixed number of iterations τ, increase of training or validation errors. In order to produce posterior class probabilities as the ensemble outputs, we apply (8). The proposed ensemble technique is summarized in Algorithm 1.

[7]We derive β_i w.r.t $\mathbf{F}'(t)$ since initially $\beta_i = 0$ and hence $\mathbf{F}'(t+1) = \mathbf{F}'(t)$.

Algorithm 1 ECB Algorithm With RBFN

Input: training set X.
Calculate initial pseudo-labels $\hat{y}_{ni}$ as Equation (4).
Generate a subset S according to Algorithm 2.
Fit an initial RBFN $f_{ni} = \text{softmax}(z_{ni})$ to $\hat{y}_{ni}$, $n \in S$.
Assign initial ensemble $\mathbf{Z}'(0) = \mathbf{Z}$.
for $t = 0$ to $\tau - 1$ **do**
  for $j = 1$ to c **do**
    Find residuals r_{ni} w.r.t. $\mathbf{F}'$ with Equation (9).
    Calculate new label estimates $\hat{y}_{nj} = \text{softmax}(r_{nj})$.
    Generate S according to Algorithm 2.
    Fit a RBFN f_{ni} to $\hat{y}_{ni}$, where $n \in S$.
    Find multiplier $\boldsymbol{\beta}$ using Equation (10).
    Update ensemble linear combination $\mathbf{Z}'(t+1)$ with Equation (11).
  end for
  Update the posterior class probabilities with $f'_{ni}(t+1) = \text{softmax}(z'_{nj}(t+1))$
end for
Output: posterior class probabilities $\mathbf{F}'$.

F. Cluster-Based Subset Selection

In order to use RBFNs as base classifiers, it is necessary to select the basis (centers) for the hidden layer. The number of basis affects the inversion of the hessian matrix in (10). Assigning all available data as centers can be prohibitive for large data sets. Therefore, sampling instances can alleviate the time complexity of the training of individual learners and improve the RBFN combination. This can be accomplished by restricting the number of basis and reducing the size of the training set of each base classifier. Therefore, the centers of each RBFN coincide with the selected instances S at each boosting iteration.

An uniform random selection of basis can cover a large proportion of the available data even when a small fraction of the training set is sampled in each iteration [10]. However, using a small sample of the available unlabeled data may not generalize as well as employing the entire data set [1, Ch. 18]. Uniform sampling may not select particular regions in space, which leads to poor estimates, i.e., (2), in these regions. It can also choose uninformative points. For example, instances far from the decision boundary are the ones with the most certain label estimates, whereas the region near the decision boundary is the one where the classifier is most likely to produce wrong labels. Thus, it is useful to select as many points from that region and very few instances that are far from that boundary.

The method in [29] is a fast forward selection algorithm that chooses instances with the highest absolute values in their current residuals, which incurs only a minor time cost. However, this method might not be reliable on SSC, as uncertain instances might also have low residuals, which may cause these instances to not be selected.

Linearly dependent points are considered redundant in [30] and should be discarded. Only independent instances may contain useful information. Their algorithm attempts to select informative instances by searching for approximately linearly independent points. However, this procedure is unsupervised and, as in uniform sampling, may choose points that have very certain labels, which do not contribute to the improvement of the decision boundary.

We propose a semisupervised cluster-based selection procedure that greedily picks only the most useful instances for the training set S of each base learner. At each boosting iteration, we form an initial set $D = L$. In order to cover as much of the information available as possible, we sample d unlabeled instances drawn from each cluster according to its probability distribution in $\mathbf{Q}$ and add them to the set D, which will have $d + l$ instances. The sum $\sum_{j \in V_n} p(\mathbf{q}_j, \mathbf{q}_n)$ is an estimate of the similarity between point n and its neighbors. If such a sum is less than a threshold θ, n will be regarded as an outlier and will not be in S.

We select a subset S of size s from D of instances with the lowest difference between the two highest-scoring classes in label estimates $\hat{\mathbf{y}}_n$, as these points are the ones typically closest to the current ensemble's decision boundary and more likely to have information on the correct shape of such a surface. Instances that do not belong to S are likely to be correctly classified independently whether they are in training set S or not. The subset S can approximate the residual gradient step at each boosting iteration. This procedure is summarized in Algorithm 2.

Algorithm 2 Basis Selection Algorithm

Input: θ, d, s, L, U, $\mathbf{Q}$, $C = \{C_1, \ldots, C_g\}$
Output: S
$D = L$
for $i = 1$ to g **do**
  $D = D \cup \{\text{i.i.d. sample of size } \frac{|C_i|d}{m} \text{ from cluster } C_i \text{ according to } \mathbf{Q}\}$
end for
for $n = 1$ to m **do**
  if $\sum_{j \in V_n} p(\mathbf{q}_j, \mathbf{q}_n) < \theta$ **then**
    $D = D \backslash \{n\}$
  end if
  $S = \{\text{The } s \text{ instances in } D \text{ with the lowest difference between the two highest-scoring labels}\}$.
end for

Such a selection costs $\mathcal{O}(dm + d \log d)$. It reduces the time complexity of the base learner from $\mathcal{O}(ec^2h^2u)$ to $\mathcal{O}(ec^2h^2s + dm + d \log d)$, where $s \ll u$. Our experiments demonstrate that ECB maintain comparable performance with state-of-the-art algorithms. Our results also show a comparison with our proposed algorithm without sampling procedure and with exact neighbor selection.

The calculation of loss function requires time of $\mathcal{O}(\tau cvu)$, where τ is the number of base learners (iterations) in ECB. Then, along with the time complexity of the base learner, ECB time complexity becomes $\mathcal{O}(ec^2h^2s + \tau(cvu + dm + d \log d))$. Therefore, unlike existing state-of-the-art ensemble methods [6], [7], [16], the time complexity of ECB grows linearly with the number of unlabeled instances.

IV. Experimental Studies

In this section, we perform experiments with two settings: transductive and inductive. We show the selection of parameters and discuss results with artificial and real-world data sets. A grid search with tenfold cross-validation was used to tune all algorithms. We also present a comparison in terms of efficiency and effectiveness with state-of-the-art algorithms using large-scale data sets.[8]

A. Methods and Parameter Tuning

Since MCSSB [6] uses all three SSC assumptions, we expect ECB to outperform MCSSB only on data sets where the cluster assumption holds, that is, a meaningful cluster structure is, in fact, present in such data. MCSSB would deliver better results on data sets where there is an unclear or no cluster structure. As its base classifier, we chose SVM, since it delivered the best results in our preliminary experiments for large-scale data sets. We fixed the parameter[9] $C = 10000$. The parameter σ was searched in $\{0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.5, 0.8, 1\}$. We set sample size with $s \in \{0.1, 0.5, 0.8, 1\}$ for transductive and inductive contexts and for large-scale data sets we fixed the sample size at 0.1. The number of base learners was set to 200 for large-scale data sets in order to have a fair comparison with ECB. And we searched it in $\{20, 50\}$ for the remainder of the experiments.

For RegBoost [7], the number of boosting iterations was set to 20 and 50 for inductive and transductive settings and 200 for the large-scale experiments. The number of neighbors was searched in $\{3, 4, 5, 6\}$. The resampling rate in the first iteration was fixed at 0.1. And the resampling rate for the remaining base classifiers was searched in $\{0.1, 0.25, 0.5\}$. For large-scale data sets, we fixed the resampling rate at 0.1. We selected SVM as base classifier following the results obtained in our preliminary experiments and [7].

In ECB, λ controls the amount of semisupervised regularization. Since the presence of a informative cluster structure is unknown, we performed a broad search for λ in the interval $[10^{-4}, 10^{0}]$ [7]. We suggest setting this value between 0 and 1, since different values might degrade generalization performance. In order to provide diversity to the ensemble, λ is uniformly drawn from the interval $[10^{-4}, 10^{0}]$ for base classifiers. Assigning the same λ value for all base classifiers did not improve the generalization error according to our preliminary experiments.

As in [9], the number of neighbors υ was fixed at 30 for most data sets used in this work, as our preliminary experiments suggested. Further tuning might improve generalization accuracy. For the approximate nearest neighbors algorithm, we fixed the target precision parameter at 0.7 and the other parameters were set as in [24].

We employed LSC to generate $\mathbf{Q}$ instantiated with Gaussian mixture models, as such method is able to find clusters with arbitrary shapes and can be employed to large-scale data sets. We fixed the number of landmarks to 200 for all settings, as our preliminary experiments suggested that it is a good tradeoff between efficiency and generalization ability. For the remainder of its configuration, we followed [23], where the selection of landmarks is performed by the k-means algorithm.

The number of clusters g should be set to, at least, the number of classes [9]. We also examined larger number of clusters (multiples of the number of classes). In the case where the clustering algorithm does not produce partitions that forms the class structure, the number of clusters can be increased, as a single class might be composed of multiple clusters. The classifier will avoid splitting these clusters and may generate a decision boundary that does not divide a particular class [9]. We searched g in the set of $\{1, 2, 3, 4\}$ times the number of classes for all experimental settings.

Our preliminary experiments suggested the grid search in the set $\{0.0001, 0.001, 0.01, 0.05, 0.1, 0.5, 0.75, 1, 1.5, 2\}$ for the tuning of the width σ. The initial subset size d was fixed at 1000. The number of selected instances s was fixed at 100. The parameter α was uniformly drawn from $[0.2, 0.5]$ for each base classifier. Ranges for λ, α, and center widths were empirically assessed in our preliminary experiments.

In transductive and inductive settings, we fixed the number of base classifier at 20, η was fixed at 0.5 and the number of IRLS iterations for RBFN was set to 50. Further optimization on these values can improve results. For large-scale, the number of base classifiers that was fixed at 200.

For CBoost, we used LSC as clustering algorithm and we calculated the exact nearest neighbors (no approximation techniques). For the rest of the parameters, we followed the tuning scheme as for ECB. In Table I, we summarize the tuning of each parameter in the compared methods.

B. Transductive Setting

In this section, we aim to establish the advantages of ECB over single classifiers and other ensembles with one or more SSC assumptions on transductive learning. In this setting, test instances are regarded as unlabeled data and generalization error is the training error on unlabeled data. Chapelle *et al.* [1] designed several transductive benchmarks. Among those, we used three artificial data sets, namely, g241c, g241d, and Digit1. And four real-world data sets: USPS, COIL, BCI, and Text.

The cluster assumption holds in g241c, that is, its classes correspond to clusters. Whereas g241d was especially built so that the cluster assumption is misleading and the manifold assumption does not hold. Digit1 was generated with a low-dimensional manifold embedded into a high-dimensional space; it does not possess a cluster structure. It is also expected that both cluster and manifold assumptions hold in USPS data set. Such data sets are summarized in Table II. Further details of these data sets can be found in [1, Ch. 21].

Each data set has 12 subsets of 10 and 100 labeled instances, and the algorithms are run 12 times with 10 and 100 labels and the mean error is reported. Tables III and IV present the results of ECB, CBoost [10], ClusterReg [9], RegBoost [7] and the algorithms in [1, Ch. 21].

[8] All data sets were standardized with zero mean and standard deviation of one.

[9] Our preliminary experiments and [6] showed that this parameter should be set to 10000. Lower and higher values did not improve the performance.

TABLE I

SUMMARY OF THE PARAMETER TUNING FOR ENSEMBLES. CLUSTERREG FOLLOWS THE PARAMETER SELECTION OF THE BASE LEARNER IN ECB

Grid search for ECB and CBoost	
λ	Grid search in $[10^{-4}, 10^{0}]$
v	30
g	$\{1, 2, 3, 4\} * c$
σ	$\{0.0001, 0.001, 0.01, 0.05, 0.1, 0.5, 0.75, 1, 1.5, 2\}$
η	0.5
α	Uniformly drawn from $[0.2, 0.7]$
d	1000
s	100
e	50
τ	20
τ for large datasets	200
Clustering algorithm	LSC with GMM
Number of LSC landmarks	200
Base learner	ClusterReg with RBFN
Nearest neighbor approximation (ECB only)	Proposed in [24]
Grid search for MCSSB	
C	10000
σ	$\{0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.5, 0.8, 1\}$
s	$\{0.1, 0.5, 0.8, 1\}$
s for large datasets	0.1
number of base learners	$\{20, 50\}$
number of base learners for large datasets	200
base learner	SVM
Grid search for RegBoost	
number of neighbors	$\{3, 4, 5, 6\}$
resampling rate in the first iteration	0.1
resampling rate	$\{0.1, 0.25, 0.5\}$
resampling rate for large datasets	0.1
number of base learners	$\{20, 50\}$
number of base learners for large datasets	200
base learner	SVM

TABLE II

SUMMARY OF DATA SETS IN TRANSDUCTIVE SETTING

Datasets	# classes	# instances	# attributes
g241c	2	1500	241
g241d	2	1500	241
Digit1	2	1500	241
USPS	2	1500	241
COIL	6	1500	241
BCI	2	400	114
Text	2	1500	11960

As expected, for the g241c data set (with 10 and 100 labeled instances), ECB was superior to all manifold-based algorithms since such a data set holds the cluster assumption. When compared to cluster-based classifiers applied to g241c with 10 labeled instances, ECB obtained better performance than most algorithms, except for the single classifier ClusterReg, which indicates that the ensemble approach might have overfit the data. With 100 labels, ECB outperformed most algorithms, except for CBoost. Such fact might indicate that the use of approximate nearest neighbors could not find informative points. Despite of g241d having a misleading cluster structure, ECB achieved comparable results with SGT (best performance) and was superior to all other cluster-based algorithms, with 10 labeled instances. With 100 labeled instances, ECB also obtained comparable results with the best algorithm (ClusterReg). Such a performance is explained by the use

TABLE III

AVERAGE OF ERRORS (%) OF RUNS WITH 12 SUBSETS OF 10 LABELED INSTANCES. FOR ALL THE ALGORITHMS, THE TEST SETS ARE FIXED. THE TABLE REPORTS ONLY THE MEAN OF THE RESULTS, AS IN [1, CH. 21]. BOLD FACE DENOTES THE BEST RESULT

Algorithm	g241c	g241d	Digit1	USPS	COIL	BCI	Text
Manifold-based algorithms							
1NN	44.05	43.22	23.47	19.82	65.91	48.74	39.44
SVM	47.32	46.66	30.60	20.03	68.36	49.85	45.37
MVU+1NN	48.68	47.28	11.92	14.88	65.72	50.24	39.40
LEM+1NN	47.47	45.34	12.04	19.14	67.96	49.94	40.48
QC+CMN	39.96	46.55	9.80	**13.61**	59.63	50.36	40.79
Discrete Reg.	49.59	49.05	12.64	16.07	63.38	49.51	40.37
SGT	22.76	**18.64**	8.92	25.36	n/a	49.59	29.02
Laplacian RLS	43.95	45.68	**5.44**	18.99	**54.54**	48.97	33.68
CHM (normed)	39.03	43.01	14.86	20.53	n/a	46.90	n/a
Cluster-based and multiple-assumptions algorithms							
TSVM	24.71	50.08	17.77	25.20	67.50	49.15	31.21
Cluster-Kernel	48.28	42.05	18.73	19.41	67.32	48.31	42.72
Data-Rep. Reg.	41.25	45.89	12.49	17.96	63.65	50.21	n/a
LDS	28.85	50.63	15.63	15.57	61.90	49.27	**27.15**
AdaBoost	40.12	43.05	28.92	25.57	71.16	47.08	47.42
ASSEMBLE	40.62	44.41	23.49	21.77	65.49	48.96	49.13
RegBoost	38.22	42.90	17.94	17.41	65.39	46.73	34.96
ClusterReg (MLP)	**16.90**	40.82	12.06	19.42	65.51	**45.36**	40.48
ClusterReg (RBFN)	26.94	27.95	10.64	19.98	69.13	49.19	40.48
CBoost	22.76	23.07	14.72	19.98	64.33	48.50	43.77
ECB	19.90	20.84	12.76	19.98	65.22	48.29	43.66

TABLE IV

AVERAGE OF ERRORS (%) OF RUNS WITH 12 SUBSETS OF 100 LABELED INSTANCES. FOR ALL THE ALGORITHMS, THE TEST SETS ARE FIXED. THE TABLE REPORTS ONLY THE MEAN OF THE RESULTS, AS IN [1, CH. 21]. BOLD FACE DENOTES THE BEST RESULT

Algorithm	g241c	g241d	Digit1	USPS	COIL	BCI	Text
Manifold-based algorithms							
1NN	40.28	37.49	6.12	7.64	23.27	44.83	30.77
SVM	23.11	24.64	5.53	9.75	22.93	34.31	26.45
MVU+1NN	44.05	43.21	3.99	6.09	32.27	47.42	30.74
LEM+1NN	42.14	39.43	2.52	6.09	36.49	48.64	30.92
QC+CMN	22.05	28.20	3.15	6.36	10.03	46.22	25.71
Discrete Reg.	43.65	41.65	2.77	**4.68**	**9.61**	47.67	24.00
SGT	17.41	9.11	2.61	6.80	n/a	45.03	**23.09**
Laplacian RLS	24.36	26.46	2.92	**4.68**	11.92	31.36	23.57
CHM (normed)	24.82	25.67	3.79	7.65	n/a	36.03	n/a
Cluster-based and multiple-assumptions algorithms							
TSVM	18.46	22.42	6.15	9.77	25.80	33.25	24.52
Cluster-Kernel	13.49	4.95	3.79	9.68	21.99	35.17	24.38
Data-Rep. Reg.	20.31	32.82	**2.44**	5.10	11.46	47.47	n/a
LDS	18.04	28.74	3.46	4.96	13.72	43.97	23.15
AdaBoost	24.82	26.97	9.09	9.68	22.96	24.02	26.31
ASSEMBLE	27.19	27.42	6.71	8.12	21.84	28.75	27.77
RegBoost	20.54	23.56	4.58	6.31	21.78	**23.69**	23.25
ClusterReg (MLP)	13.38	**4.36**	3.45	5.25	24.73	33.92	32.09
ClusterReg (RBFN)	19.54	17.07	7.20	16.53	36.35	48.11	32.09
CBoost	**12.71**	6.99	4.34	7.20	30.67	38.83	25.58
ECB	15.12	7.27	4.65	7.25	30.15	38.86	25.63

of cluster neighborhood. In such a method, classes can be represented by more than one cluster and, even though the data distribution does not match the class distribution, these classes can be identified by several clusters. Therefore, classifiers that use this technique, ClusterReg and CBoost, could overcome such a misleading structure.

Manifold-based algorithms are expected to deliver better generalization in Digit1 data set [1]. This expectation was confirmed on data sets with 10 labeled instances. With 100 labeled instances, ECB obtained better generalization ability than other cluster-based techniques due to its robustness to the uncertain labeled points in clusters. Data-dependent regularization, ClusterReg, and CBoost produced better predictive accuracy than ECB, which can be explained by ECB's approximate neighborhoods.

Both cluster and manifold assumptions are expected to hold in USPS data set [1]. However, with 10 and 100 labeled instances, manifold-based algorithms delivered best performance (QC+CMN, discrete regularization and laplacian RLS). ECB obtained comparable accuracy with cluster-based methods for both amounts of labeled data. Such results might indicate that, in fact, the manifold present in the data is relevant for classification.

TABLE V
SUMMARY OF DATA SETS IN INDUCTIVE SETTING

Datasets	# classes	# instances	# attributes
Australian	2	690	16
Balance scale	3	625	6
Bupa	2	345	8
Contraceptive	3	1473	11
Dermatology	6	366	36
Ecoli	5	327	8
German	2	1000	26
Glass	6	214	11
Haberman	2	306	5
Heart cleveland	5	303	15
Horse colic	2	368	28
House votes	2	435	18
Ionosphere	2	351	35
Mammographic masses	2	961	7
Pima indians diabetes	2	768	10
SPECT	2	267	24
Statlog (V. silhouettes)	4	846	20
Transfusion	2	748	6
WDBC	2	569	32
Yeast	9	1479	10

The structures of COIL, BCI, and Text data sets are unknown. Nonetheless, ECB yielded competitive performance among cluster-based and manifold-based algorithms on these real-world data sets with 10 and 100 labeled instances. This might denote ECB's robustness to uncertain unlabeled points and scarce labels.

C. Inductive Setting

In inductive learning, classifiers predict labels of unseen instances. We use this setting to evaluate ECB along with state-of-the-art algorithms: ClusterReg, CBoost, MCSSB, and RegBoost. We selected 20 data sets from the UCI machine learning repository [31]. Table V summarizes the data sets employed.

We generated three versions of each data set, the proportion of labeled data $\frac{l}{m}$ in each variant is 5%, 10%, and 20%. We transformed these data sets into semisupervised problems by randomly selecting the respective amount of labeled instances for each data set. The labeled instances of each data set are different for each version, so that each data set variant poses a different problem.

In order to improve error estimation, all labels in test set were available. It is not possible to know in advance the true class structure and the corresponding SSC assumption that these real-world data sets possess. Then, the success of a classifier will depend on the proper matching between their assumptions and the actual class structure of the data [1]. Ensemble-based algorithms with multiple assumptions may deliver higher average performance throughout various data sets [7], that is, such methods are more likely to deliver better predictions than a specialist algorithm that implements the wrong assumption for a given data set. In this sense, we compare ECB to two ensemble classifiers that use all assumptions—MCSSB and RegBoost—and a cluster-based ensemble, CBoost.

Table VI(a)–(c) shows the means and standard deviations of the generalization errors of all algorithms for all data sets with 5%, 10%, and 20% of labeled data, respectively. We employ a pairwise t-test with 95% of significance level. Symbols $\bullet/\circ$ indicate whether ECB is statistically superior/inferior and Win/Tie/Loss denotes the number of data sets where ECB is significantly superior/comparable/inferior to the compared algorithm. The Friedman test [32] with 5% of significance provided statistical evidence of the difference between the means of errors in Table VI. After the Friedman test, we performed the Bonferroni–Dunn test [32] with 5% of significance level. Such a posthoc test confirmed that ECB was superior to all other algorithms, including state-of-the-art methods, across all amounts of labeled data.

We used only real-world data sets (Table V) with unknown structures. When compared to MCSSB and RegBoost, ECB was statistically superior in most data sets across all amounts of labeled instances, as shown in Table VI. This fact indicates that our method was robust to fewer labeled instances and potentially overlapping classes. The decision boundary generated by ECB might have not been severely affected by the position of labeled points in dense regions. Such a performance can be explained by the use of semisupervised base learners. Such base learners are able to seek the neighborhood for an appropriate label of an unlabeled instance and might recover from incorrect label estimates.

Both MCSSB and RegBoost implement all three semisupervised assumptions. In cases where there is a clear cluster structure, the quality of the decision boundary generated by these algorithms might be limited by the search for a useful manifold. In such situations, methods specialized in finding informative clusters may yield significantly superior generalization, as evidenced in Table VI.

When compared to ClusterReg, ECB was able to significantly improve generalization in many data sets with 10% and 20% of labeled data. Therefore, our ensemble approach was able to recover from errors of base classifiers, despite the use of approximate nearest neighbors and sampled training sets. However, for 5% of labeled instances, ECB only statistically improved over ClusterReg in five data sets. This result might indicate that ECB's greedy approach might have overfit.

As expected, ECB obtained similar generalization ability to CBoost across the majority of data sets. Therefore, despite the use of approximation and subset selection techniques, ECB was successful in relatively small real-world data sets when compared to algorithms that compute exact nearest neighbors and uses all data available in each iteration.

D. Large-Scale Setting

In this section, we present a scalability and convergence study of ECB in comparison to other methods. We compare the computation time and generalization error of ECB to MCSSB, RegBoost, and CBoost.[10] Our experiment used six

[10]The CPU time was measured in an Intel(R) Xeon(R) CPU at 2.20GHz with 64 gigabytes of memory. All algorithms were implemented in MATLAB(R). The implementation of ECB can be further optimized.

TABLE VI

MEAN AND STANDARD DEVIATION (%) OF TENFOLD CROSS-VALIDATION ERROR AT 5%, 10%, AND 20% OF LABELED DATA. •/◦ INDICATES WHETHER ECB IS STATISTICALLY SUPERIOR/INFERIOR TO THE COMPARED METHOD, ACCORDING TO PAIRWISE T-TEST AT 95% OF SIGNIFICANCE LEVEL. WIN/TIE/LOSS DENOTES THE NUMBER OF DATA SETS WHERE ECB IS SIGNIFICANTLY SUPERIOR/COMPARABLE/INFERIOR TO THE COMPARED ALGORITHM. (a) RESULTS FOR 5% OF LABELED DATA. (b) RESULTS FOR 10% OF LABELED DATA. (c) RESULTS FOR 20% OF LABELED DATA

Datasets	MCSSB	RegBoost	ClusterReg	CBoost	ECB
Australian credit	44.52 ± 4.87 ◦	18.15 ± 3.74	41.88 ± 17.14 ◦	18.67 ± 1.27 ◦	20.03 ± 1.46
Balance scale	26.06 ± 5.58 •	57.30 ± 11.24 •	9.82 ± 1.90	15.98 ± 4.93 •	11.84 ± 3.53
Bupa	38.91 ± 10.85	47.45 ± 10.83	30.50 ± 2.21 ◦	38.90 ± 4.90	41.33 ± 6.99
Contraceptive	57.07 ± 4.59 •	67.76 ± 8.20 •	49.85 ± 1.27	52.74 ± 2.84 •	49.53 ± 2.93
Dermatology	11.12 ± 5.82	58.24 ± 5.63 •	23.39 ± 7.44 •	5.34 ± 5.31	7.03 ± 5.36
Ecoli	18.66 ± 5.96 •	37.62 ± 6.83 •	16.54 ± 4.74 •	11.68 ± 2.81	11.73 ± 2.62
German credit	31.46 ± 5.59 •	52.62 ± 21.26 •	23.27 ± 1.91	29.03 ± 2.61 •	25.25 ± 2.58
Glass	60.31 ± 10.60	77.53 ± 17.87 •	58.40 ± 9.29	36.31 ± 10.36 ◦	57.33 ± 11.01
Haberman	33.15 ± 11.00	31.53 ± 17.19	16.91 ± 3.06 ◦	29.09 ± 2.71 ◦	35.35 ± 3.26
Heart cleveland	47.34 ± 15.06	61.06 ± 7.89 •	40.85 ± 3.53 ◦	53.42 ± 3.79 •	45.71 ± 6.22
Horse colic	30.38 ± 10.08	48.44 ± 19.57 •	31.06 ± 5.61	26.23 ± 6.17	27.34 ± 6.33
House votes	61.57 ± 7.24 •	56.10 ± 12.64 •	7.81 ± 3.06	7.84 ± 2.30	7.58 ± 2.10
Ionosphere	35.64 ± 12.78 •	50.55 ± 19.80 •	12.97 ± 2.51 •	13.35 ± 7.78	11.15 ± 1.46
Mammographic masses	46.34 ± 4.80 •	25.42 ± 4.94 •	12.73 ± 3.19	15.24 ± 3.36 •	11.96 ± 2.79
Pima indians diabetes	34.82 ± 4.62 •	34.21 ± 7.51 •	27.05 ± 1.96	29.66 ± 2.86 •	26.76 ± 2.53
SPECT	79.51 ± 10.71 •	31.99 ± 4.27 •	11.09 ± 1.78	11.08 ± 3.12	9.89 ± 2.82
Vehicle silhouettes	49.47 ± 6.09 •	69.71 ± 5.89 •	52.11 ± 5.51 •	35.33 ± 5.59	34.41 ± 4.34
Transfusion	23.88 ± 6.03 •	34.59 ± 23.03 •	19.65 ± 6.21	29.46 ± 5.48 •	19.07 ± 6.28
WDBC	37.25 ± 5.37 •	18.93 ± 5.67 •	8.69 ± 1.17	6.86 ± 2.68 ◦	8.69 ± 0.74
Yeast	56.58 ± 3.03 •	68.63 ± 3.68 •	53.35 ± 2.12 •	48.78 ± 0.99	48.22 ± 1.82
Win/Tie/Loss	13/6/1	17/3/0	5/11/4	7/9/4	–

(a) Results for 5% of labeled data.

Australian credit	44.58 ± 6.90 •	13.38 ± 2.54 ◦	12.76 ± 1.46 ◦	16.18 ± 2.73	15.64 ± 2.37
Balance scale	23.40 ± 5.29 •	46.80 ± 9.48 •	5.47 ± 2.69	4.45 ± 1.72	4.60 ± 1.43
Bupa	43.64 ± 9.92 •	47.11 ± 12.00 •	33.22 ± 2.43 •	23.55 ± 4.31	24.85 ± 5.26
Contraceptive	53.35 ± 3.51 •	61.00 ± 4.59 •	45.71 ± 1.91	46.65 ± 1.80 •	43.83 ± 3.55
Dermatology	9.97 ± 6.31 •	69.25 ± 5.95 •	20.12 ± 6.85 •	1.26 ± 1.64	1.33 ± 2.36
Ecoli	18.59 ± 6.63	35.11 ± 7.51 •	18.90 ± 5.93 •	19.17 ± 4.57 •	14.71 ± 2.71
German credit	32.35 ± 5.22 •	48.28 ± 16.27 •	22.55 ± 1.54 •	22.83 ± 3.67	21.01 ± 1.87
Glass	52.54 ± 11.18 •	67.30 ± 12.24 •	43.09 ± 9.44 •	19.54 ± 4.36	19.48 ± 3.00
Haberman	42.59 ± 10.20 •	29.91 ± 10.65	22.64 ± 5.44 ◦	34.62 ± 5.88	32.88 ± 2.68
Heart cleveland	52.73 ± 11.12 •	72.12 ± 12.89 •	37.81 ± 2.34	48.32 ± 3.71 •	40.43 ± 4.78
Horse colic	25.35 ± 9.32	57.12 ± 18.39 •	30.10 ± 6.89 •	22.52 ± 5.19	23.11 ± 4.21
House votes	61.35 ± 8.08 •	58.12 ± 11.63 •	11.76 ± 1.24 •	1.78 ± 1.23 ◦	6.39 ± 3.34
Ionosphere	35.90 ± 6.75 •	44.85 ± 15.40 •	10.48 ± 2.08	8.27 ± 2.20	8.54 ± 3.80
Mammographic masses	46.21 ± 6.15 •	21.11 ± 2.72 •	12.26 ± 1.50	23.02 ± 4.94 •	12.61 ± 1.87
Pima indians diabetes	34.84 ± 6.50 •	32.75 ± 5.10 •	27.90 ± 2.30 •	28.39 ± 3.34 •	24.22 ± 2.80
SPECT	79.60 ± 8.61 •	49.55 ± 32.36 •	15.45 ± 1.49 •	11.70 ± 1.58	12.12 ± 1.06
Vehicle silhouettes	43.46 ± 7.23 •	74.44 ± 2.74 •	55.63 ± 3.75 •	37.90 ± 2.31	36.33 ± 3.34
Transfusion	23.79 ± 6.93 •	35.07 ± 7.15 •	15.87 ± 1.94	26.77 ± 16.47	17.73 ± 3.23
WDBC	37.37 ± 7.19 •	13.86 ± 6.47 •	2.77 ± 1.49	5.31 ± 2.05 •	2.81 ± 2.12
Yeast	53.90 ± 3.70 •	68.63 ± 2.94 •	52.09 ± 3.50 •	47.57 ± 1.66	46.43 ± 1.50
Win/Tie/Loss	18/2/0	18/1/1	11/7/2	6/13/1	–

(b) Results for 10% of labeled data.

Australian credit	44.34 ± 7.04 •	17.37 ± 5.21	16.14 ± 3.12 ◦	15.82 ± 3.44 ◦	18.52 ± 2.68
Balance scale	23.85 ± 8.12 •	55.06 ± 5.23 •	3.45 ± 1.08	2.62 ± 2.45	3.47 ± 1.79
Bupa	38.25 ± 10.96 •	52.16 ± 11.77 •	20.41 ± 5.00	21.22 ± 4.65	18.10 ± 6.09
Contraceptive	54.15 ± 6.38 •	57.22 ± 6.41 •	45.80 ± 3.09	43.52 ± 2.12	44.86 ± 3.58
Dermatology	6.52 ± 3.99 •	59.61 ± 8.20 •	14.71 ± 4.92 •	4.13 ± 2.24	3.46 ± 1.77
Ecoli	17.59 ± 7.73	37.52 ± 13.14 •	18.37 ± 3.34	12.93 ± 7.42 ◦	19.06 ± 6.43
German credit	33.83 ± 6.82 •	37.56 ± 16.99 •	23.90 ± 2.73 •	19.73 ± 2.34	20.85 ± 2.87
Glass	61.69 ± 12.82 •	67.06 ± 9.44 •	19.42 ± 6.93	20.32 ± 7.66 •	14.40 ± 6.16
Haberman	32.57 ± 9.23 •	25.95 ± 7.57 •	17.47 ± 5.46	25.60 ± 7.41	19.98 ± 7.60
Heart cleveland	52.30 ± 12.83 •	56.29 ± 16.76 •	41.09 ± 6.02	39.83 ± 5.19	36.69 ± 5.98
Horse colic	40.87 ± 10.84 •	47.22 ± 14.98 •	37.05 ± 3.09 •	29.12 ± 5.04	29.17 ± 4.75
House votes	61.29 ± 7.43 •	50.04 ± 10.84 •	6.87 ± 2.88 •	3.11 ± 1.99	4.30 ± 2.11
Ionosphere	36.03 ± 10.85 •	38.46 ± 13.65 •	8.59 ± 1.78	8.59 ± 1.78	7.93 ± 2.31
Mammographic masses	46.45 ± 4.85 •	46.73 ± 5.50 •	10.52 ± 1.72	10.36 ± 2.33	10.94 ± 2.13
Pima indians diabetes	34.88 ± 7.24 •	31.74 ± 5.47 •	22.98 ± 3.42	26.55 ± 2.69 •	22.26 ± 2.92
SPECT	79.53 ± 5.20 •	30.85 ± 12.03 •	8.07 ± 2.53	8.23 ± 4.06	7.64 ± 2.77
Vehicle silhouettes	33.47 ± 4.32	72.05 ± 5.28 •	50.83 ± 5.46 •	34.26 ± 4.72	30.81 ± 4.48
Transfusion	23.78 ± 5.44 •	26.61 ± 4.43 •	16.63 ± 2.24	20.81 ± 3.69	18.18 ± 3.14
WDBC	37.28 ± 6.42 •	28.99 ± 5.33 •	1.32 ± 1.14	1.97 ± 1.28 •	0.89 ± 1.15
Yeast	52.47 ± 4.27 •	68.65 ± 2.65 •	51.35 ± 2.79 •	46.57 ± 2.61	46.64 ± 3.12
Win/Tie/Loss	18/2/0	19/1/0	6/13/1	3/15/2	–

(c) Results for 20% of labeled data.

large real-world data sets, which are summarized in Table VII. We randomly selected 100 labeled instances for each data set.

In Fig. 2, we show the generalization ability and efficiency of ECB along with MCSSB, RegBoost, and CBoost across different amounts of unlabeled data. As depicted in Fig. 2(a),

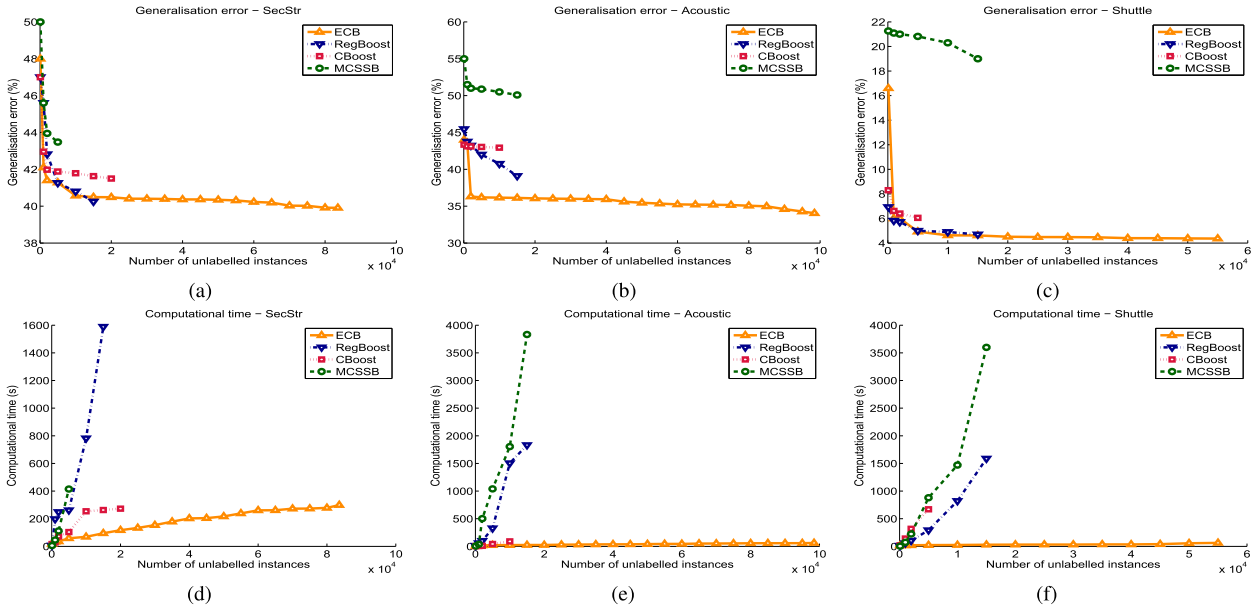


Fig. 2. Plots of generalization error and computational time versus the increase of the number of unlabeled instances on SecStr, Acoustic, and Shuttle data sets. Points used in one run are also employed in the next runs. (a) and (d) Generalization error and computational time for the SecStr data set. (b) and (e) Generalization error and computational time for the Acoustic data set. (c) and (f) Generalization error and computational time for the Shuttle data set.

TABLE VII

SUMMARY OF LARGE DATA SETS. MCSSB, REGBOOST, AND CBOOST COULD NOT HANDLE THE HIGHLIGHTED DATA SETS

Datasets	# classes	# instances	# attributes
SecStr	2	83,679	315
Acoustic	3	98,528	50
Shuttle	7	58,000	9
Covtype	7	581,012	54
Hepmass	2	10,500,000	27
SUSY	2	5,000,000	18

all methods reduce their test error with the increase of the number of unlabeled instances, which might denote the usefulness of unlabeled data.

Existing algorithms were not able to handle the data sets highlighted in Table VII. MCSSB, RegBoost, and CBoost fail with a few thousands of instances, as shown in Fig. 2(d). MCSSB updates each instance weight with the consideration of all other unlabeled points, that is, it uses all instances to assign the pseudolabel of an unlabeled instance. This leads to a quadratic growth of computational time with respect to the number of unlabeled points. Moreover, MCSSB stores a $m \times m$ similarity matrix. Such facts cause the algorithm to fail with a limited computational budget.

RegBoost requires the computation of exact nearest neighbors, which involves the use of a $m \times m$ distance matrix. As indicated in Fig. 2(d)–(f), such an algorithm has large space complexity with relatively small amounts of data, which also leads to an increase in CPU time. Therefore, likewise MCSSB, with a certain small number of instances, RegBoost fails due to unfeasible running time and memory consumption.

In Fig. 2(b) and (e), the algorithms reduce their generalization error with larger amounts of unlabeled data. However, as depicted in Fig. 2(e), only ECB was able to handle full data sets. In Fig. 2(c), MCSSB did not deliver comparable accuracy with the other algorithms, this fact may indicate that its decision boundary was affected by the search for manifolds when clusters are relevant. Likewise the Acoustic data set, only ECB showed scalability for Shuttle data set [Fig. 2(f)].

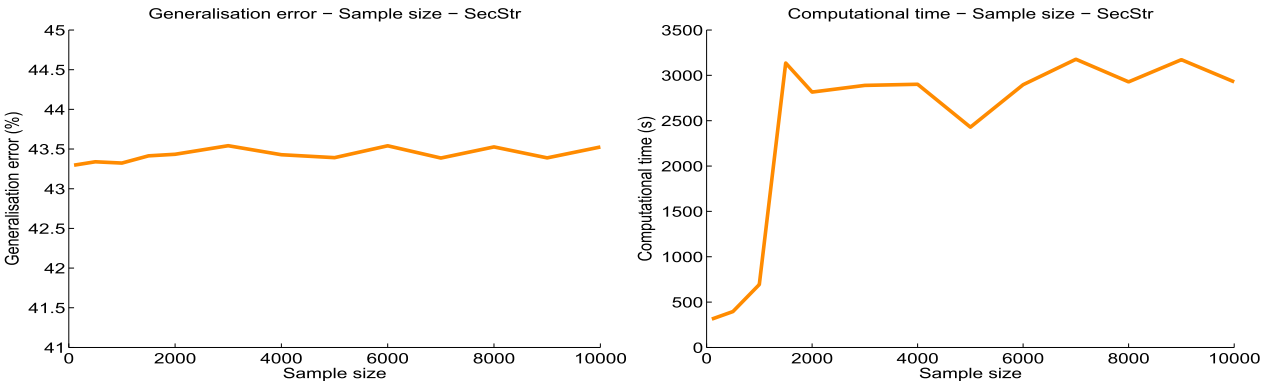
As shown in Fig. 2(d)–(f), the time requirement of ECB grows linearly with the number of unlabeled instances. And, as depicted in Fig. 2(a)–(c), ECB can also produce comparable results with existing algorithms.

In this sense, the employed clustering algorithm, LSC, is suitable for large data sets, delivering good partitions efficiently without compromising memory usage. The approximation technique increases efficiency in terms of both time and memory, which solves the drawbacks from RegBoost and MCSSB with respect to high memory consumption (such drawbacks also have an impact on execution time due to the overhead caused by virtual memory). And the subset selection also greatly reduces time complexity and allows training with large data sets in reasonable time.

E. Subset Selection

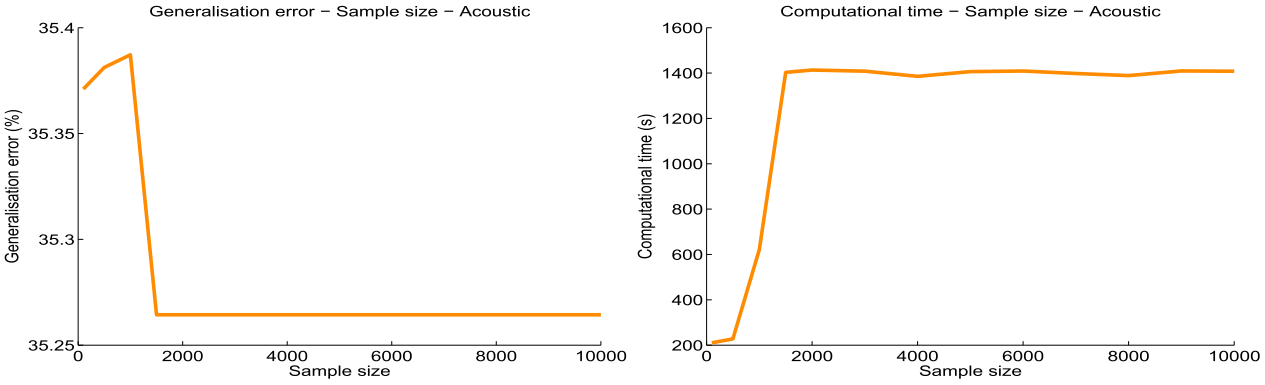
In this section, we assess the selection procedure and the size of S with respect to accuracy and efficiency. Figs. 3 and 4 show the impact of different sample sizes s on generalization error and computational time. Since ECB was the only method that could handle the highlighted data sets in Table VII, we use these data sets to evaluate different subset selection procedures for the proposed algorithm.

In order to evaluate the sensitivity of ECB to the sample size s regarding accuracy and efficiency, Figs. 3 and 4 present the generalization error and CPU time on SecStr and Acoustic data sets, respectively, for different amounts of sampled data. The amount of unlabeled data needed on SecStr is small,



(a) (b)

Fig. 3. Analysis of the influence of the amount of sampled points s on (a) generalization error and (b) computational time of ECB on SecStr data set.



(a) (b)

Fig. 4. Analysis of the influence of the amount of sampled points s on (a) generalization error and (b) computational time of ECB for Acoustic data set.

as shown in Fig. 3(a), which denotes that such a data set does not possess a clear cluster structure. Hence, labeled data will be more important for the training algorithm. In contrast, ECB could improve its generalization ability with larger amounts of sampled data for Acoustic data set [Fig. 4(a)].

As shown in Figs. 3(a) and 4(b), the computational time stabilizes when the sample size reaches the limit of the number of hidden nodes in the RBFN. This behavior is expected since the number of centers employed in RBFN increases with s until it reaches a threshold (in this case, 2000 hidden nodes).[11]

We also plotted the generalization error throughout 1000 iterations (number of base learners) of ECB on SecStr, Acoustic, and Shuttle data sets [Fig. 5(a)–(c), respectively] without a second termination criterion. We verified that the proposed algorithm converges with a small number of base learners, despite the small number of sampled instances, $s = 100$, in each iteration. And, as expected, the algorithm starts to overfit in later iterations. The ensemble tends to overfit the training data as gradient boosting is a greedy approach. Such a figure suggests that ECB can be successfully employed to large-scale data sets with a large number of base learners without compromising time efficiency.

[11]The complexity of RBFN grows quadratically with the number of centers and we limit such a parameter.

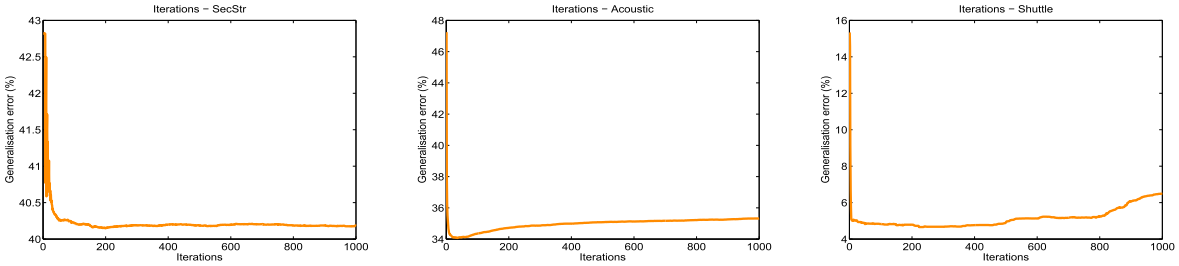
TABLE VIII

GENERALIZATION ERROR ± STANDARD DEVIATION OF DIFFERENT SUBSET SELECTION PROCEDURES. BOLD FACE INDICATES STATISTICALLY SUPERIOR PERFORMANCE

Datasets	Random	Kernel-based	Cluster-based
Covtype	45.65 ± 2.72	39.60 ± 1.46	**38.17 ± 1.28**
Hepmass	11.80 ± 0.68	**10.38 ± 0.1**	**10.4 ± 0.13**
SUSY	27.92 ± 0.41	25.23 ± 0.64	**24.61 ± 0.60**

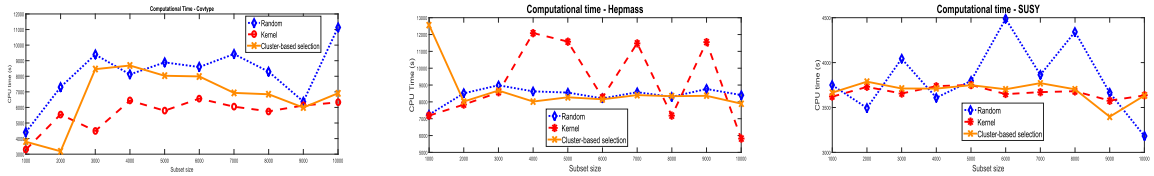
In order to evaluate the convergence of ECB, we plotted its generalization error in Fig. 5. We fixed the maximum number of iterations (number of base learners) at 1000 and implemented no other stopping criterion in order to assess the impact of number of base learners on training and overfitting. We also compared the generalization error and computational time of three different subset selection approaches: random, kernel-based [30], and the proposed greedy cluster-based selection. In Table VIII, we present the generalization ability of these three methods in the data sets highlighted in Table VII.

In order to study the predictive performance and efficiency of the proposed subset selection method, we compare the greedy cluster-based procedure to random and kernel-based selection. In Table VIII, the proposed method outperforms all other algorithms according to pairwise t-test. Fig. 6 shows the CPU time of ECB with the three different subset selection



(a) (b) (c)

Fig. 5. Plots of iterations (number of base learners) versus generalization errors of ECB on (a) SecStr, (b) Acoustic, and (c) Shuttle data sets, respectively.



(a) (b) (c)

Fig. 6. Computational time versus subset size of ECB with different subset selection methods on (a) covtype, (b) hepmass, and (c) SUSY data sets, respectively.

TABLE IX
TIME AND MEMORY COMPLEXITIES, $s, d \ll u$

Datasets	Time	Memory
MCSSB	$\mathcal{O}(\tau cu^2 + \tau s^3)$	$\mathcal{O}(m^2)$
RegBoost	$\mathcal{O}(vcm \log m + c\tau s^3 + \tau cvu)$	$\mathcal{O}(m^2)$
CBoost	$\mathcal{O}(ec^2h^2u + \tau cvu)$	$\mathcal{O}(m^2)$
ECB	$\mathcal{O}(ec^2h^2s + \tau(cvu + dm + d \log d))$	$\mathcal{O}(vu)$

methods on the data sets in bold face in Table VII. Subset sizes are varied from 10^3 to 10^4. According to Fig. 6, our selection method also presents comparable time efficiency to the other approaches.

Table IX summarizes time and memory complexities of the methods employed in our experiments. The computational time of ECB does not grow quickly (exponentially) with the number of sampled instances. ECB is able to tackle large data sets with a wide range of sampled instances. In the analysis with these figures, we used the validation error as a stopping criterion. The slight instability is due to early stopping (termination criterion) caused by the increase of validation error. In the greedy cluster-based selection, the decision of selecting each point depends on the density of its region and on its relative position (estimated by label uncertainty) to the decision boundary. Unlike the compared unsupervised methods, our algorithm was able to select relevant unlabeled points for the training set of each base learner, which causes a better estimated correction of the current decision surface. In fact, our experiments confirmed that the proposed method is suitable for large-scale data sets.

V. CONCLUSION

In this paper, we introduced an ECB algorithm. State-of-the-art algorithms cannot be applied to large data sets due to their time and memory requirements. The proposed method employs an efficient clustering algorithm, approximates nearest neighbors and a greedy cluster-based selection to reduce the training set of base learners. Such improvements reduce both time and memory requirements of ECB.

We designed three experimental settings: transductive, inductive, and large-scale data sets. In both transductive and inductive settings, ECB could deliver comparable predictive performance to state-of-the-art algorithms. In our analysis on large-scale data sets, we evaluated and validated the scalability of the proposed algorithm.

The experimental analysis confirmed the following.

1) The use of our selection method along with approximate nearest neighbors and LSC can increase the efficiency of ECB and maintains comparable generalization performance with other methods.
2) ECB is robust to the position of labeled data in a cluster.
3) By using semisupervised base learners, it is robust to incorrect label assignments during training.

ECB depends on the presence of relevant clusters. We intend to study the use of the manifold assumption in ECB in order to propose an efficient multiassumption method to big data problems.

REFERENCES

[1] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.
[2] T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Norwell, MA: Kluwer, 2002.
[3] M. Szummer and T. Jaakkola, "Partially labeled classification with Markov random walks," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2002, pp. 945–952.
[4] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CALD-02-107, 2002.
[5] M. H. Nguyen, H. A. Abbass, and R. I. Mckay, "A novel mixture of experts model based on cooperative coevolution," *Neurocomputing*, vol. 70, nos. 1–3, pp. 155–163, 2006.
[6] H. Valizadegan, R. Jin, and A. K. Jain, "Semi-supervised boosting for multi-class classification," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases-II (ECMLPKDD)*, 2008, pp. 522–537.

[7] 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.

[8] A. Saffari, C. Leistner, and H. Bischof, "Regularized multi-class semi-supervised boosting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 967–974.

[9] 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.

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

[11] R. G. F. Soares, "Cluster-based semi-supervised ensemble learning," Ph.D. dissertation, School Comput. Sci., Univ. Birmingham, Birmingham, U.K., Jul. 2014. [Online]. Available: http://etheses.bham.ac.uk/4818/

[12] G. S. Mann and A. McCallum, "Simple, robust, scalable semi-supervised learning via expectation regularization," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 593–600. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273571

[13] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 290–297.

[14] X. Zhu, "Semi-supervised learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1530, 2005.

[15] V. Sindhwani and S. S. Keerthi, "Large scale semi-supervised linear SVMs," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2006, pp. 477–484. [Online]. Available: http://doi.acm.org/10.1145/1148170.1148253

[16] G. Yu, G. Zhang, Z. Yu, C. Domeniconi, J. You, and G. Han, "Semi-supervised ensemble classification in subspaces," *Appl. Soft Comput.*, vol. 12, no. 5, pp. 1511–1522, 2012.

[17] E. Song, D. Huang, G. Ma, and C.-C. Hung, "Semi-supervised multi-class adaboost by exploiting unlabeled data," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 6720–6726, 2011.

[18] 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.

[19] O. Delalleau, Y. Bengio, and N. L. Roux, "Large-scale algorithms," in *Semi-Supervised Learning*, O. Chapelle, B. Schölkopf, and A. Zien, Eds. Cambridge, MA, USA: MIT Press, 2006, ch. 18.

[20] X. Zhu and J. Lafferty, "Harmonic mixtures: Combining mixture models and graph-based methods for inductive and scalable semi-supervised learning," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 1052–1059. [Online]. Available: http://doi.acm.org/10.1145/1102351.1102484

[21] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[22] P. Sun and X. Yao, "Sparse approximation through boosting for learning large scale kernel machines," *IEEE Trans. Neural Netw.*, vol. 21, no. 6, pp. 883–894, Jun. 2010.

[23] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation," in *Proc. 25th AAAI Conf. Artificial Intell.*, 2011, pp. 313–318.

[24] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. Int. Conf. Comput. Vis. Theory Appl. (VISSAPP)*, 2009, pp. 331–340.

[25] I. T. Nabney, "Efficient training of RBF networks for classification," in *Proc. 9th Int. Conf. Artif. Neural Netw. (ICANN)*, vol. 1. Sep. 1999, pp. 210–215.

[26] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.

[27] L. Z. Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2004, pp. 1601–1608.

[28] P. M. Vaidya, "An $O(n \log n)$ algorithm for the all-nearest-neighbors problem," *Discrete Comput. Geometry*, vol. 4, no. 2, pp. 101–115, 1989. [Online]. Available: http://dx.doi.org/10.1007/BF02187718

[29] P. B. Nair, A. Choudhury, and A. J. Keane, "Some greedy learning algorithms for sparse regression and classification with mercer kernels," *J. Mach. Learn. Res.*, vol. 3, pp. 781–801, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944954

[30] 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.

[31] A. Frank and A. Asuncion. (2010). "UCI machine learning repository." [Online]. Available: http://archive.ics.uci.edu/ml

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

Rodrigo G. F. Soares received the B.Sc. degree in computer engineering from the Federal University of Rio Grande do Norte, Natal, Brazil, in 2005, the M.Phil. degree in computer science from the Federal University of Pernambuco, Recife, Brazil, in 2008, and the Ph.D. degree in computer science from the University of Birmingham, Birmingham, U.K., in 2014, with a scholarship from Capes Foundation, Brazil.

He is currently a Lecturer with the Federal Rural University of Pernambuco, Recife. His current research interests include data mining, semisupervised learning, neural networks, clustering, and evolutionary computation.

Dr. Soares was a recipient of Brazilian Council for Scientific and Technological Development (CNPq) scholarships in both degrees.

Huanhuan Chen (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 Professor with the UBRI Laboratory, School of Computer Science and Technology, USTC. His current research interests include statistical machine learning, data mining, fault diagnoses, and evolutionary computation.

Prof. Chen was a recipient of the 2015 International Neural Network Society Young Investigator Award, the IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation Award (the only winner), the 2009 CPHC/British Computer Society Distinguished Dissertations Award (the runner up), and the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award (bestowed in 2011 and only one paper in 2009) for his work on probabilistic classification vector machines on Bayesian machine learning. 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.

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, Haidian, China, in 1985, and the Ph.D. degree from USTC in 1990.

He is currently the Chair Professor of computer science with the Southern University of Science and Technology, Shenzhen, China, and a Professor of computer science with the University of Birmingham, Birmingham, U.K. He has been researching multiobjective optimization since 2003, when he published a well-cited EMO03 paper on many objective optimization. His current research interests include evolutionary computation, ensemble learning, and their applications in software engineering.

Prof. Yao was a recipient of the 2001 IEEE Donald G. Fink Prize Paper Award, the 2010, 2016, and 2017 IEEE Transactions on Evolutionary Computation Outstanding Paper Awards, the 2010 BT 1213 Gordon Radley Award for the Best Author of Innovation (finalist), the 2011 IEEE Transactions on Neural Networks Outstanding Paper Award, the prestigious Royal Society Wolfson Research Merit Award in 2012, the IEEE Computational Intelligence Society (CIS) Evolutionary Computation Pioneer Award in 2013, and many other best paper awards. He was the President of the IEEE CIS from 2014 to 2015, and the Editor-in-Chief of the IEEE Transactions on Evolutionary Computation from 2003 to 2008. He is a Distinguished Lecturer of the IEEE CIS.