Semi-Supervised Regression with Co-Training Style Algorithms

Zhi-Hua Zhou, *Senior Member, IEEE,* and Ming Li

Abstract

The traditional setting of supervised learning requires a large amount of labeled training examples in order to achieve good generalization. However, in many practical applications, unlabeled training examples are readily available but labeled ones are fairly expensive to obtain. Therefore, semi-supervised learning has attracted much attention. Previous research on semi-supervised learning mainly focuses on semi-supervised classification. Although regression is almost as important as classification, semi-supervised regression is largely understudied. In particular, although *co-training* is a main paradigm in semi-supervised learning, few works has been devoted to co-training style semi-supervised regression algorithms. In this paper, a co-training style semi-supervised regression algorithm, i.e. COREG, is proposed. This algorithm uses two regressors each labels the unlabeled data for the other regressor, where the confidence in labeling an unlabeled example is estimated through the amount of reduction in mean square error over the labeled neighborhood of that example. Analysis and experiments show that COREG can effectively exploit unlabeled data to improve regression estimates.

Index Terms

Machine Learning; Data Mining; Learning with Unlabeled Data; Semi-Supervised Learning; Semi-Supervised Regression; Co-Training

Manuscript received August 13, 2006; revised January 21, 2007; accepted July 31, 2007. This work was supported by the National Science Foundation of China (60325207, 60635030), the Jiangsu Science Foundation of China (BK2004001), the Foundation for the Author of National Excellent Doctoral Dissertation of China (200343), and the National High Technology Research and Development Program of China (2007AA01Z169).

The authors are with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China (e-mail: {zhouzh, lim}@lamda.nju.edu.cn).

I. INTRODUCTION

In the traditional setting of supervised learning, a large amount of training examples should be available in order to construct a model with good generalization ability. It is noteworthy that these training examples should be *labeled*, that is, the labels of these training examples should be known in advance. Unfortunately, in many practical machine learning and data mining applications, although a large number of unlabeled training examples can be at hand, usually only a few labeled training examples are available because obtaining the labels requires human effort. For example, in web user profile analysis, it is easy to get a lot of web user profiles but assigning labels such as *profitable user* or *non-profitable user* to these data requires the inspection, judgement, or even time-consuming tracing by human assessors, which is fairly expensive. Therefore, exploiting unlabeled data to help supervised learning has become a hot topic during the past few years.

Currently there are mainly three machine learning paradigms for exploiting unlabeled examples, that is, semi-supervised learning, transductive learning and active learning. Semi-supervised learning [11], [54] deals with methods which attempt to automatically exploit unlabeled examples where the unlabeled examples are usually different from the test examples; transductive learning [40], [23] deals with methods which also attempt to automatically exploit unlabeled examples but assuming that the unlabeled examples are exactly the test examples; active learning [1], [34] deals with methods which assume the learner has some control over the input space, and an *oracle* can be queried for labels of specific instances, with the goal of minimizing the number of queries required. In this paper, semi-supervised learning is considered.

Many developments have been achieved in the research on semi-supervised learning. However, it is noteworthy that previous research mainly focuses on classification. Although regression is almost as important as classification, semi-supervised regression remains largely understudied. In particular, *co-training* [8] has been recognized as one of the main paradigms of semi-supervised learning, but its usefulness in semi-supervised regression has not been investigated well. In this paper, a co-training style semi-supervised regression algorithm named COREG, i.e. CO-training REGressors, is proposed. This algorithm employs two regressors each of which labels the unlabeled data for the other during the learning process. In order to choose appropriate unlabeled examples to label, COREG estimates the labeling confidence by consulting the influence of the labeling of unlabeled examples on the labeled examples. The final prediction is made by combining the regression estimates generated by both regressors. Note that COREG seeks the diversity between regressors through using different distance metrics and/or number of neighbors instead of requiring *two views* of the data, and thus it is applicable to regression problems with no natural attribute partitions. Analysis and experiments show that this algorithm can effectively exploit unlabeled data to improve regression estimates.

The rest of this paper is organized as follows. Section 2 briefly reviews semi-supervised learning. Section 3 proposes the COREG algorithm. Section 4 presents an analysis on the algorithm. Section 5 reports on the experiments. Finally, Section 6 concludes.

II. SEMI-SUPERVISED LEARNING

The research on semi-supervised learning is usually dated back to Shahshahani and Landgrebe's work [35], but the usefulness of unlabeled examples in supervised learning has actually been recognized earlier [25]. A likely reason for that there has been few works on this problem during the early years is that it seems difficult to incorporate unlabeled data directly within conventional supervised learning methods such as Backpropagation neural networks [26]. With the rapid progress of machine learning, especially the explosive bloom of statistical learning research, and the increasing requirement of exploiting unlabeled data, semi-supervised learning has become a hot topic in both machine learning and data mining.

There are many effective semi-supervised learning approaches. Roughly speaking, most of these approaches can be categorized into three main paradigms. In the first paradigm, a generative model such as Naïve Bayes classifier or mixture of Gaussians is used for the classifier, and EM [16] is employed to model the label estimation or parameter estimation process. Representative approaches of this paradigm include [19], [26], [28], [35]. In the second paradigm, unlabeled data is used to regularize the learning process in various ways. For example, a graph can be defined on the data set, where the nodes correspond to the labeled or unlabeled examples while the edges reflect the similarity between the examples; then, the label smoothness can be enforced over the graph as a regularization term. Representative approaches of this paradigm include [4], [5], [7], [46], [55]. The third paradigm, i.e. co-training [8], is closely related to the work described in this paper, therefore here we introduce it with more details.

The co-training method proposed by Blum and Mitchell [8] trains two classifiers separately on two *sufficient and redundant views*, i.e. two attribute sets each of which is sufficient for learning and conditionally independent to the other given the class label, and uses the predictions of each classifier on unlabeled examples to augment the training set of the other.

Such an idea of utilizing the natural redundancy in the attributes has been employed in some other works. For example, Yarowsky [45] performed word sense disambiguation by constructing a sense classifier using the local context of the word and a classifier based on the senses of other occurrences of that word in the same document; Riloff and Jones [32] classified a noun phrase for geographic locations by considering both the noun phrase itself and the linguistic context in which the noun phrase appears; Collins and Singer [12] performed named entity classification using both the spelling of the entity itself and the context in which the entity occurs; etc.

Dasgupta et al. [15] have theoretically shown that when the requirement of sufficient and redundant views is met, the co-trained classifiers could make few generalization errors by maximizing their agreement over the unlabeled data. As Nigam and Ghani [27] reported, when an independent and redundant attribute split exists, the co-training algorithm outperforms many other semi-supervised learning algorithms in utilizing unlabeled data; even when there is no natural attribute divisions, if there are sufficient redundancy among the attributes and a fairly reasonable division of the attributes can be identified, then the co-training algorithm may show similar advantages to other algorithms.

However, although co-training has been used in many domains such as statistical parsing and noun phrase identification [22], [29], [33], [38], in most scenarios the requirement of sufficient and redundant views, or even the requirement of sufficient redundancy, could not be met. Therefore, researchers attempt to develop variants of the co-training algorithm for relaxing such a requirement.

Goldman and Zhou [20] proposed an algorithm which does not exploit attribute partition. This algorithm requires using two different supervised learning algorithms that partition the instance space into a set of equivalence classes, and employs cross validation to determine how to label the unlabeled examples and how to produce the final hypothesis. Zhou and Li [51] proposed the *tri-training* approach, which uses three classifiers such that the *labeling confidence* (i.e., how confident a classifier is in labeling an unlabeled example) can be implicitly obtained through consulting the agreement of the classifiers. By contrast, such labeling confidence should be explicitly measured in previous co-training algorithms when a classifier attempts to label examples to the other classifier or when the classifications made by different classifiers are to be merged. This algorithm does not require attribute partition, nor does it require using different types of learning algorithms. Moreover, since more classifiers are involved, it is possible to exploit ensemble learning [17] to help improve generalization. Recently this method has been extended to use more learners to make better use of the power of ensemble learning, which achieves success in computer-aided medical diagnosis [24]. Another co-training style algorithm which uses more than two learners has been presented by Zhou and Goldman [47]. Some variants of co-training [48], [49] which combine semi-supervised learning with active learning and do not require different views, have been applied to content-based image retrieval, where images provided by the user in the query and relevance feedbacks are regarded as labeled examples while the images existing in the image database are regarded as unlabeled examples.

Note that Balcan et al. [3] have theoretically shown that given appropriately strong PAC-learners on each view, an assumption of *expansion* on the underlying data distribution, which is weaker than the assumption of sufficient and redundant views, is sufficient for *iterative co-training* to succeed. This implies that the *conditional independence* [8] or even the *weak dependence* [2] between the two views is not needed, at least, for iterative co-training which is actually the working routine taken by many co-training style algorithms [20], [47], [48], [49], [51]. In fact, the assumption of two sufficient views is too strong that Zhou et al. [53] have shown that when this assumption can be met, semi-supervised learning given only one labeled example is feasible. Recently, Wang and Zhou [43] have theoretically shown that co-training style algorithms can be effective if the learners are diverse, which implies that the two views is actually used to achieve the diversity of the learners, and therefore they are not needed if the diversity can be achieved from other channels.

As mentioned before, previous research on semi-supervised learning mainly study semi-supervised classification. Although regression is in general as important as classification, only a few studies have been devoted to semi-supervised regression. One reason for this fact is that the popular *cluster assumption* (i.e., similar instances should have the same label) in semi-supervised classification does not naturally hold for regression problems and therefore most semi-supervised classification methods are not straightforward applicable to regression. Fortunately, another well-known assumption, the *manifold assumption* (i.e., similar instances should have similar labels), still holds in regression problems, and thus, by exploiting the local smoothness in the feature space, semi-supervised regression is feasible. In addition to our work [50], Brefeld et al. [9] developed another co-training style semi-supervised regression algorithm, CORLSR, which extends a technique used in semi-supervised classification [36]. Semi-supervised kernel [37] has also been studied in regression tasks [18], [30], [42].

III. COREG

Let $L = \{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_{|L|}, \mathbf{y}_{|L|})\}$ denote the labeled example set, where $\mathbf{x}_i$ is the i-th instance described by d attributes, $\mathbf{y}_i$ is its real-valued label, i.e. its expected real-valued output, and $|L|$ is the number of labeled examples; let U denote the unlabeled data set, where the instances are also described by the d attributes, whose real-valued labels are unknown, and $|U|$ is the number of unlabeled examples.

Two regressors, i.e. h_1 and h_2, are generated from L, each of which is then refined with the help of unlabeled examples that are labeled by the latest version of the other regressor. Here the kNN regressor [14] is used as the base learner to instantiate h_1 and h_2, which labels a new instance through averaging the real-valued labels of its k-nearest neighboring examples. The use of kNN regressor is due to the following considerations. First, in semi-supervised learning, the regressors will be refined in each of many learning iterations. Since kNN is a lazy learning method which does not hold a separate training phase, the refinement of the kNN regressors can be more efficiently realized than that of regressors such as neural networks which require a separate training phase. Second, in order to choose appropriate unlabeled examples to label, the labeling confidence should be estimated. Since the manifold assumption of local smoothness holds in regression problems, in COREG the estimation utilizes the neighboring properties of the training examples, which can be easily coupled with kNN regressors.

It is noteworthy that according to [43], in order to launch an effective co-training process, the initial learners must be diverse. Extremely, if they are identical, then for either regressor, the unlabeled examples labeled by the other regressor may be the same as these labeled by the regressor itself. Consequently, the algorithm degenerates to *self-training* [27] with a single learner. In the standard setting of co-training, the use of sufficient and redundant views enables the learners to be different. Previous research has also shown that even when there is no natural attribute partitions, a fairly reasonable attribute partition will enable co-training to exhibit advantages if there are sufficient redundancy among the attributes [27]. While in an extended co-training algorithm which does not require sufficient and redundant views [20], the diversity among the learners is achieved through using different learning algorithms. Since COREG assumes neither two views nor different learning algorithms, the diversity of the regressors has to be sought from other channels.

Here the diversity is achieved by utilizing different distance metrics and/or different k values. In fact, two key points of a kNN learner are how to identify the nearest neighbors and how many nearest neighbors are considered for a given instance. By using different distance metrics, the vicinities identified for a given instance can be different even when the same k value is used; while, by using different k values, the predictions for a given instance can also be different even when the same distance metric is used. Thus, the kNN regressors h_1 and h_2 can be diverse by instantiating them with different distance metrics and/or different k values. Such a setting can also bring another advantage, that is, since it is usually difficult to decide which distance metric and which k value is better for the concerned task, the functions of these regressors may be somewhat complementary if they are combined. Note that the use of different distance metrics has been shown helpful in some variants of co-training [48], [49].

In order to choose appropriate unlabeled examples to label, the labeling confidence should be estimated such that the most confidently labeled example can be identified. Note that both active learning and semi-supervised learning try to select "valued" unlabeled examples to use. In active learning, the selected unlabeled example will be passed to an *oracle* to ask for its ground-truth label. Therefore, the unlabeled example on which the learner is with the least confidence is usually selected since it would be most valuable for improving the learner. While in semi-supervised learning, since there is no *oracle* that can be relied, the unlabeled example on which the learner is with the most confidence is usually selected to label. These two learning processes have been combined in some previous work [48], [49].

Estimating the labeling confidence in classification is relatively straightforward because when making classifications, many classifiers can also provide an estimated probability (or an approximation) for the classification, e.g. a Naïve Bayes classifier returns the maximum a posteriori hypothesis where the posterior probabilities can be used; a Backpropagation neural network classifier returns thresholded classification where the real-valued outputs can be used; etc. Therefore, the labeling confidence can be estimated through consulting the probabilities of the unlabeled examples being labeled to different classes. For example, suppose the probability of the instance

$\mathbf{a}$ being classified to the classes c_1 and c_2 is 0.90 and 0.10, respectively, while that of the instance $\mathbf{b}$ is 0.60 and 0.40, respectively. Then the instance $\mathbf{a}$ is more confident to be labeled (to class c_1). Unfortunately, in regression there is no such estimated probability that can be used directly. This is because in contrast to classification where the number of class labels to be predicted is finite, the possible predictions in regression are infinite. Therefore, a key of COREG is the mechanism for estimating the labeling confidence.

Intuitively, the most confidently labeled example of a regressor should be with such a property, i.e. the error of the regressor on the labeled example set should decrease the most if the most confidently labeled example is utilized. In other words, the most confidently labeled example should be the one which makes the regressor most *consistent* with the labeled example set. Thus, the mean squared error (MSE) of the regressor on the labeled example set can be evaluated first. Then, the MSE of the regressor utilizing the information provided by $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ can be evaluated on the labeled example set, where $\mathbf{x}_u$ is an unlabeled instance while $\hat{\mathbf{y}}_u$ is the real-valued label generated by the original regressor. Let Δ_u denote the result of subtracting the latter MSE from the former MSE. Note that the number of Δ_u to be estimated equals to the number of unlabeled examples. Finally, $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ associated with the biggest positive Δ_u can be regarded as the most confidently labeled example.

Since repeatedly measuring the MSE of the kNN regressor on the whole labeled example set in each iteration will be time-consuming, considering that kNN regressor mainly utilizes local information, COREG employs an approximation. That is, for each $\mathbf{x}_u$, COREG identifies its k-nearest labeled examples and uses them to compute the MSE. In detail, for each $\mathbf{x}_u$, let Ω_u denote the set of its k-nearest neighbors in L, then the most confidently labeled example $\tilde{\mathbf{x}}$ is identified through maximizing the value of $\delta_{\mathbf{x}_u}$ in Eq. 1,

$$\delta_{\mathbf{x}_u} = \sum_{\mathbf{x}_i \in \Omega_u} \left(\left(\mathbf{y}_i - h\left(\mathbf{x}_i\right)\right)^2 - \left(\mathbf{y}_i - h'\left(\mathbf{x}_i\right)\right)^2 \right) \tag{1}$$

where h denotes the original regressor while h' denotes the refined regressor which has utilized the information provided by $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$, $\hat{\mathbf{y}}_u = h(\mathbf{x}_u)$.

The pseudo code of COREG is shown in Table I, where the function $kNN(L_j, k_j, D_j)$ returns a kNN regressor on the labeled example set L_j, whose k value is k_j and distance metric is D_j. The learning process stops when the maximum number of learning iterations, T, is reached, or there is no unlabeled example which is capable of reducing the MSE of any of the regressors on

TABLE I

PSEUDO-CODE DESCRIBING THE COREG ALGORITHM

```
ALGORITHM: COREG
INPUT: labeled example set $L$, unlabeled example set $U$,
       maximum number of learning iterations $T$,
       number of nearest neighbors $k_1, k_2$
       distance metrics $D_1, D_2$
PROCESS:
  $L_1 \leftarrow L$; $L_2 \leftarrow L$
  Create pool $U'$ of size $s$ by randomly picking examples from $U$
  $h_1 \leftarrow kNN(L_1, k_1, D_1)$; $h_2 \leftarrow kNN(L_2, k_2, D_2)$
  Repeat for $T$ rounds:
    for $j \in \{1, 2\}$ do
      for each $\mathbf{x}_u \in U'$ do
        $\Omega_u \leftarrow Neighbors\,(\mathbf{x}_u, L_j, k_j, D_j)$
        $\hat{\mathbf{y}}_u \leftarrow h_j(\mathbf{x}_u)$
        $h'_j \leftarrow kNN(L_j \cup \{(\mathbf{x}_u, \hat{\mathbf{y}}_u)\}, k_j, D_j)$
        $\delta_{\mathbf{x}_u} \leftarrow \sum_{\mathbf{x}_i \in \Omega_u} \left( (\mathbf{y}_i - h_j\,(\mathbf{x}_i))^2 - \left(\mathbf{y}_i - h'_j\,(\mathbf{x}_i)\right)^2 \right)$
      end of for
      if there exists an $\delta_{\mathbf{x}_u} > 0$
      then $\tilde{\mathbf{x}}_j \leftarrow \arg\max_{\mathbf{x}_u \in U'} \delta_{\mathbf{x}_u}$; $\tilde{\mathbf{y}}_j \leftarrow h_j(\tilde{\mathbf{x}}_j)$
           $\pi_j \leftarrow \{(\tilde{\mathbf{x}}_j, \tilde{\mathbf{y}}_j)\}$; $U' \leftarrow U' - \{\tilde{\mathbf{x}}_j\}$
      else $\pi_j \leftarrow \emptyset$
    end of for
    $L_1 \leftarrow L_1 \cup \pi_2$; $L_2 \leftarrow L_2 \cup \pi_1$
    if neither of $L_1$ and $L_2$ changes then exit
    else
      $h_1 \leftarrow kNN(L_1, k_1, D_1)$; $h_2 \leftarrow kNN(L_2, k_2, D_2)$
      Replenish $U'$ to size $s$ by randomly picking examples from $U$
  end of Repeat
  $f_1 \leftarrow Regressor(L_1)$; $f_2 \leftarrow Regressor(L_2)$
OUTPUT: regressor $f^*(\mathbf{x}) \leftarrow \frac{1}{2}\,(f_1\,(\mathbf{x}) + f_2\,(\mathbf{x}))$
```

the labeled example set. According to Blum and Mitchell's suggestion [8], a pool of unlabeled examples smaller than U is used, and the iterative co-training routine is executed. Note that in each iteration the unlabeled example chosen by h_1 won't be chosen by h_2, which is an extra mechanism for encouraging the diversity of the regressors. Thus, even when h_1 and h_2 are similar, the examples they label for each other will still be different.

In each iteration the computational cost of COREG is mainly spent on identifying the neighbors of examples. Since the neighboring labeled examples for every labeled training example can be computed and stored in advance, actually only the neighborhood of the unlabeled examples need to be identified and then the neighbors of the labeled training examples could be updated. Moreover, the identified neighborhood of many unlabeled examples can be reused in iterations. So, the computational cost of COREG is almost comparable to that of using kNN regressors to predict the unlabeled examples.

Note that after using two kNN regressors to select and label the unlabeled examples, the predictions can be made by other kinds of regressors instead of the kNN regressors. For example, suppose we use linear regression. After using the two kNN regressors to select and label the unlabeled examples, we get two augmented labeled training sets. On each training set we can train a linear regressor. Then, the predictions of these two linear regressors are averaged as the final prediction.

IV. ANALYSIS

This section attempts to analyze whether the learning process of COREG can use the unlabeled examples to improve the regression estimates. In order to simplify the discussion, here the effect of the pool U' is not considered as in [8]. That is, the unlabeled examples are assumed as being picked from the unlabeled example set U directly.

In each learning iteration of COREG, for each unlabeled example $\mathbf{x}_u$, its k-nearest neighboring labeled examples are put into the set Ω_u. As mentioned before, the newly labeled example should make the regressor become more consistent with the labeled data set. Therefore, the goodness of $\mathbf{x}_u$ can be evaluated using a criterion shown in Eq. 2,

$$\Delta_u = \frac{1}{|L|}\sum_{\mathbf{x}_i \in L} \left(\mathbf{y}_i - h(\mathbf{x}_i)\right)^2 - \frac{1}{|L|}\sum_{\mathbf{x}_i \in L} \left(\mathbf{y}_i - h'(\mathbf{x}_i)\right)^2 \tag{2}$$

where h is the original regressor while h' is the one refined with $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$. If the value of Δ_u is positive, then utilizing $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ is beneficial.

In the COREG algorithm, the unlabeled example which maximizes the value of $\delta_{\mathbf{x}_u}$ is picked out to be labeled. Therefore, the question is, whether the unlabeled example chosen according to the maximization of $\delta_{\mathbf{x}_u}$ will result in a positive Δ_u value or not.

First, assume that $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ is among the k-nearest neighbors of some examples in Ω_u, and is not among the k-nearest neighbors of any other examples in L. In this case, it is obvious that utilizing $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ will only change the regression estimates on the examples in Ω_u, therefore Eq. 2 becomes Eq. 3.

$$\Delta_u = \frac{1}{k} \sum_{\mathbf{x}_i \in \Omega_u} (\mathbf{y}_i - h(\mathbf{x}_i))^2 - \frac{1}{k} \sum_{\mathbf{x}_i \in \Omega_u} (\mathbf{y}_i - h'(\mathbf{x}_i))^2 \tag{3}$$

Comparing Eqs. 1 with 3 it can be found that the maximization of $\delta_{\mathbf{x}_u}$ also results in the maximization of Δ_u.

Second, assume that $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ is not among the k-nearest neighbors of any example in Ω_u. In this case, the value of $\delta_{\mathbf{x}_u}$ is zero, therefore $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ won't be chosen in COREG.

Third, assume that $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ is among the k-nearest neighbors of some examples in Ω_u as well as some examples in $L-\Omega_u$, and assume that these examples in $L-\Omega_u$ are $(\mathbf{x}_1^{'}, \mathbf{y}_1^{'}), \cdots, (\mathbf{x}_m^{'}, \mathbf{y}_m^{'})$. Then Eq. 2 becomes Eq. 4.

$$\begin{aligned}\Delta_u \;\; = \;\; & \frac{1}{k+m}[\sum_{\mathbf{x}_i \in \Omega_u} \left((\mathbf{y}_i - h(\mathbf{x}_i))^2 - (\mathbf{y}_i - h'(\mathbf{x}_i))^2\right) \\ & + \sum_{q \in \{1,\cdots,m\}} \left(\left(\mathbf{y}_q^{'} - h(\mathbf{x}_q^{'})\right)^2 - \left(\mathbf{y}_q^{'} - h'(\mathbf{x}_q^{'})\right)^2\right)]\end{aligned} \tag{4}$$

Maximizing $\delta_{\mathbf{x}_u}$ will maximize the first sum term of Eq. 4, but whether it can enable Δ_u be positive should also refer the second sum term. Unfortunately, the value of this sum term is difficult to measure except that the neighboring relationships between all the labeled examples and $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ are evaluated. Therefore, there may exist cases where the unlabeled example chosen according to the maximization of $\delta_{\mathbf{x}_u}$ may decrease Δ_u, which is the cost COREG takes for using $\delta_{\mathbf{x}_u}$ that can be more efficiently computed to approximate Δ_u. Nevertheless, experiments show that in most cases such an approximation is effective.

It seems that using only one regressor to label the unlabeled examples for itself might be feasible, where the unlabeled examples can be chosen according to the maximization of $\delta_{\mathbf{x}_u}$. While considering that the labeled example set usually contains noise, the use of two regressors can be helpful to reduce overfitting. Let Λ denote the set of noisy examples in L. For the unlabeled instance $\mathbf{x}_u$, either of the regressors h_1 and h_2 will identify a set of k-nearest neighboring labeled examples for $\mathbf{x}_u$. Let $\Omega_{u,1}$ and $\Omega_{u,2}$ denote these sets, respectively. Since h_1 and h_2 use different distance metrics and/or different k values, $\Omega_{u,1}$ and $\Omega_{u,2}$ are usually different, and therefore $\Omega_{u,1} \cap \Lambda$ and $\Omega_{u,2} \cap \Lambda$ are also different. Suppose $\mathbf{x}_u$ is labeled by h_1 and then $(\mathbf{x}_u, h_1(\mathbf{x}_u))$ is put into L_1, where $h_1(\mathbf{x}_u)$ suffers from the noisy examples in $\Omega_{u,1} \cap \Lambda$. For another unlabeled instance $\mathbf{x}_v$ which is very close to $\mathbf{x}_u$, its k-nearest neighbors identified by h_1 will be very similar to $\Omega_{u,1}$ except that $(\mathbf{x}_u, h_1(\mathbf{x}_u))$ has replaced a previous neighbor. Thus, $h_1(\mathbf{x}_v)$ will be roughly affected by $(\Omega_{u,1} \cap \Lambda) \cup \{(\mathbf{x}_u, h_1(\mathbf{x}_u))\}$. Since $(\mathbf{x}_u, h_1(\mathbf{x}_u))$ has already suffered from the noisy examples in $\Omega_{u,1} \cap \Lambda$, $h_1(\mathbf{x}_v)$ will suffer from $\Omega_{u,1} \cap \Lambda$ more seriously than $h_1(\mathbf{x}_u)$ does. While, if the instance $\mathbf{x}_u$ is labeled by h_2 and $(\mathbf{x}_u, h_2(\mathbf{x}_u))$ is put into L_1, then $h_1(\mathbf{x}_v)$ will suffer from $\Omega_{u,1} \cap \Lambda$ only once, although $\mathbf{x}_u$ is still very close to $\mathbf{x}_v$. Quantitative analysis on such effect is rather difficult since it is related to the concrete data distribution, which is an interesting issue for future study.

V. Experiments

A. Configuration

Fifteen data sets are used in our experiments, including synthetic as well as real-world data sets. The synthetic data sets are tabulated in Table II. The data sets *2-d Mexican Hat* and *3-d Mexican Hat* have been used by Weston et al. [44] in investigating the performance of support vector machines; *Friedman #1*, *#2*, *#3* have been used by Breiman [10] in testing the performance of Bagging; *Gabor*, *Multi* and *SinC* have been used by Hansen [21] in comparing ensemble learning methods; *Plane* has been used by Ridgeway et al. [31] in exploring the performance of boosted naive Bayesian regressors; all these data sets have been used by Zhou et al. [52] in investigating selective ensemble. In the experiments, the instances contained in these data sets were generated from the functions listed in Table II. The constraints on the attributes are also shown in the table, where $\mathbf{U}[a, b]$ means a uniform distribution over the interval determined by a and b. Note that the input attributes as well as the real-valued labels have been normalized to $[0.0, 1.0]$. Gaussian noise terms have been added to the functions. The real-world data sets are from the UCI machine learning repository [6] and StatLib [41], as shown in Table III.

TABLE II

THE SYNTHETIC DATA SETS

Data set	Function	Attribute	Size
2-d Mexican Hat	$y = \mathrm{sinc}\,\lvert x\rvert = \frac{\sin\lvert x\rvert}{\lvert x\rvert}$	$x \sim \mathbf{U}\,[-2\pi, 2\pi]$	5,000
3-d Mexican Hat	$y = \mathrm{sinc}\sqrt{x_1^2 + x_2^2} = \frac{\sin\sqrt{x_1^2+x_2^2}}{\sqrt{x_1^2+x_2^2}}$	$x_1, x_2 \sim \mathbf{U}\,[-4\pi, 4\pi]$	3,000
Friedman #1	$y = 10\sin(\pi x_1 x_2) + 20\,(x_3 - 0.5)^2 + 10x_4 + 5x_5$	$x_1, x_2, x_3, x_4, x_5 \sim \mathbf{U}\,[0, 1]$	5,000
Friedman #2	$y = \sqrt{x_1^2 + \left(x_2 x_3 - \left(\frac{1}{x_2 x_4}\right)\right)^2}$	$x_1 \sim \mathbf{U}\,[0, 100]$ $x_2 \sim \mathbf{U}\,[40\pi, 560\pi]$ $x_3 \sim \mathbf{U}\,[0, 1]$ $x_4 \sim \mathbf{U}\,[1, 11]$	5,000
Friedman #3	$y = \tan^{-1}\frac{x_2 x_3 - \frac{1}{x_2 x_4}}{x_1}$	$x_1 \sim \mathbf{U}\,[0, 100]$ $x_2 \sim \mathbf{U}\,[40\pi, 560\pi]$ $x_3 \sim \mathbf{U}\,[0, 1]$ $x_4 \sim \mathbf{U}\,[1, 11]$	3,000
Gabor	$y = \frac{\pi}{2}\exp\left[-2\left(x_1^2 + x_2^2\right)\right]\cos\left[2\pi\,(x_1 + x_2)\right]$	$x_1, x_2 \sim \mathbf{U}\,[0, 1]$	3,000
Multi	$y = 0.79 + 1.27x_1x_2 + 1.56x_1x_4 + 3.42x_2x_5 + 2.06x_3x_4x_5$	$x_1, x_2, x_3, x_4, x_5 \sim \mathbf{U}\,[0, 1]$	4,000
Plane	$y = 0.6x_1 + 0.3x_2$	$x_1, x_2 \sim \mathbf{U}\,[0, 1]$	1,000
Polynomial	$y = 1 + 2x + 3x^2 + 4x^3 + 5x^4$	$x \sim \mathbf{U}\,[0, 1]$	3,000
SinC	$y = \frac{\sin(x)}{x}$	$x \sim \mathbf{U}\,[0, 2\pi]$	3,000

TABLE III

THE REAL-WORLD DATA SETS

Data sets	# Features	Size	Source
chscase.census6	6	400	StatLib
kin8nm_2000	8	2000	UCI
no2	7	500	StatLib
pollen	5	3848	StatLib
puma8NH_2000	8	2000	UCI

Each data set is randomly partitioned into labeled/unlabeled/test data sets according to certain ratios. Specifically, about 25% of the data are kept as test examples while the remaining 75% of the data are used as the set of training examples, i.e. $L \cup U$. In each training set, L and U are partitioned under different *label rates* including 10%, 30% and 50%. For instance, assuming a training set contains 1,000 examples, when the label rate is 10%, 100 examples are put into L

with their labels while the remaining 900 examples are put into U without their labels.

A popular routine in evaluating semi-supervised algorithms [8], [20], [27], [51], [55] is adopted. In detail, one hundred runs of experiments are conducted on each data set; in each run, algorithms are evaluated on randomly partitioned labeled/unlabeled/test splits; the average MSE at each iteration is recorded. In the experiments the maximum number of iterations, T, is set to 100, and the size of the pool used in the learning process is fixed to 100. Note that the learning process may stop before the maximum number of iterations is reached, and in that case, the final MSE is used in computing the average MSE of the following iterations.

B. Experiments on Using kNN Regressors on Synthetic Data Sets

1) Comparison with initial regression estimates: As mentioned before, COREG achieves the diversity of the two kNN regressors by employing different distance metrics and/or different k values. In our experiments, Euclidean distance and Mahalanobis distance are considered, and the k values are fixed on 3 or 5. In this section, three different parameter settings of COREG are evaluated: 1) $k_1 = k_2 = 3$, $D_1 = Euclidean$ and $D_2 = Mahalanobis$, 2) $k_1 = 3$, $k_2 = 5$ and $D_1 = D_2 = Mahalanobis$, and 3) $k_1 = 5$, $k_2 = 3$, $D_1 = Euclidean$ and $D_2 = Mahalanobis$.

The improvements on MSE obtained by exploiting unlabeled examples under different label rates are tabulated in Table IV, which is computed by subtracting the final MSE (i.e. the MSE of regressors after semi-supervised learning process) from the initial MSE (i.e. the MSE of regressors before utilizing any unlabeled examples) and then divided by the initial MSE. Pairwise t-tests with significance level 0.05 are executed and the table entries with significant improvements are boldfaced. The corresponding p-values are shown in Table V.

The tables show that COREG almost always perform significantly better than its initial regression estimates, which verifies that COREG is able to exploit the unlabeled data to improve the regression estimates on all the evaluated configurations.

Moreover, it can be found from Table IV that COREG performs better under *config-1* than under *config-2*. This is not difficult to understand since the neighborhood identified by using a smaller k is always a subset of that identified by using larger k, and thus, using different distance metrics could be more effective than using different k values in achieving the diversity of the kNN regressors. Moreover, it can be observed that COREG performs better under *config-1* than under *config-3*. One possible explanation is that although either the use of different metrics or

TABLE IV

IMPROVEMENTS (%) ON MEAN SQUARED ERROR

Data set	label rate 10%			label rate 30%			label rate 50%		
	config-1	*config-2*	*config-3*	*config-1*	*config-2*	*config-3*	*config-1*	*config-2*	*config-3*
2-d MexicanHat	**34.7**	**31.0**	**26.9**	**19.9**	**14.0**	**12.5**	**14.0**	**9.5**	**9.9**
3-d MexicanHat	**11.6**	**8.5**	**4.3**	**11.2**	**4.9**	**3.7**	**9.4**	**3.6**	**3.5**
Friedman #1	**3.0**	-0.1	-0.6	**2.5**	-0.1	-0.3	**1.7**	-0.1	0.0
Friedman #2	**7.0**	**1.3**	**1.6**	**4.7**	**0.9**	**0.9**	**3.6**	**0.7**	**1.1**
Friedman #3	**2.1**	-1.3	-1.1	**1.6**	-1.3	-0.7	**1.6**	-1.0	-0.2
Gabor	**17.6**	**15.0**	**5.7**	**14.9**	**8.2**	**7.9**	**11.3**	**6.3**	**6.4**
Multi	**5.5**	**1.4**	0.8	**4.2**	**0.8**	0.5	**3.4**	**0.7**	**0.7**
Plane	-1.4	-3.2	-2.9	0.2	-1.4	-1.3	0.3	-1.0	-1.0
Polynomial	**39.3**	**41.2**	**26.2**	**27.2**	**23.7**	**15.7**	**18.8**	**12.9**	**10.9**
SinC	**44.9**	**43.3**	**36.1**	**26.3**	**20.2**	**18.2**	**19.8**	**14.8**	**14.0**
avg.	16.4	13.7	9.7	11.3	7.0	5.7	8.4	4.6	4.5

TABLE V

THE p-VALUES OF THE PAIRWISE t-TESTS

Data set	label rate 10%			label rate 30%			label rate 50%		
	config-1	*config-2*	*config-3*	*config-1*	*config-2*	*config-3*	*config-1*	*config-2*	*config-3*
MexicanHat2d	1.80E-46	1.39E-31	7.03E-30	1.99E-51	3.66E-29	2.28E-28	1.52E-48	2.21E-27	2.29E-33
MexicanHat3d	1.25E-16	2.41E-12	1.21E-04	1.51E-24	2.63E-10	6.86E-06	8.51E-20	1.68E-04	7.89E-05
Friedman #1	1.40E-16	6.25E-01	3.30E-02	4.37E-25	6.41E-01	1.03E-01	4.32E-27	4.00E-01	8.43E-01
Friedman #2	2.29E-30	3.74E-03	5.51E-04	6.70E-34	2.12E-03	3.29E-04	1.32E-33	6.13E-04	3.77E-08
Friedman #3	2.43E-04	2.27E-02	5.47E-02	2.71E-04	4.81E-04	5.71E-02	2.25E-06	1.57E-03	6.29E-01
Gabor	3.96E-22	1.49E-15	2.57E-04	1.43E-29	4.95E-11	4.18E-11	3.75E-35	1.52E-11	5.42E-14
Multi	4.16E-17	1.02E-02	1.08E-01	9.02E-28	2.09E-03	5.35E-02	2.28E-29	1.81E-03	2.82E-03
Plane	1.05E-03	5.20E-14	1.76E-11	5.83E-01	4.71E-08	3.04E-08	2.10E-01	4.68E-06	7.94E-06
Polynomial	2.86E-32	1.20E-20	1.37E-12	1.42E-29	1.24E-18	4.71E-15	5.20E-33	2.58E-16	2.18E-16
SinC	1.14E-41	1.30E-31	1.43E-25	5.13E-61	6.07E-32	2.48E-31	8.20E-52	2.54E-30	8.84E-29

different k values can enable COREG to exploit unlabeled examples, their helpful effects might counteract when they are used together, which means that some pairs of metric and k value could return good performance but some could not. Thus, in order to exert the advantage of using different metrics and different k values simultaneously, a careful study on the cooperation of metrics and k values has to be performed. For simplicity, in the following experiments we use *config-1* as the default configuration of COREG.

2) Comparison with other methods: In order to further evaluate the performance of COREG, three semi-supervised regression methods, i.e. ARTRE, SELF1 and SELF2, are developed and compared in this section.

ARTRE is a co-training style algorithm. Since the experimental data sets are with no sufficient and redundant views, here an artificial redundant view is generated by deriving new attributes from the original ones. Let $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ denote the same example in the original view and the artificial view, respectively. For data sets with only one attribute (e.g., *SinC*), the redundant attribute $\mathbf{x}^{(2)}$ is derived by $1-\mathbf{x}^{(1)}$; for data sets with $d > 1$ attributes, a matrix $A_{d\times d}$ is employed to help generate the artificial attributes in the way of $\mathbf{x}^{(2)} = A\mathbf{x}^{(1)}$. Elements of A are filled by -1, 0, or 1, randomly. In each iteration, each kNN regressor chooses the unlabeled example that maximizes the value of $\delta_{\mathbf{x}_u}$ in Eq. 1 to label for the other regressor. Euclidean distance and $k = 3$ are used. The final prediction is made by averaging the regression estimates of these two refined regressors.

SELF1 is a self-training style algorithm. This algorithm uses one kNN regressor and in each iteration, it chooses the unlabeled example which maximizes the value of $\delta_{\mathbf{x}_u}$ in Eq. 1 to label for itself. The k value and distance metric used by SELF1 are set to the same as these used by the first regressor of COREG, that is, $k = 3$ and $D = Euclidean$.

SELF2 is another self-training style algorithm. This algorithm uses two kNN regressors each employs self-training for its own refinement. The parameter setting of SELF2 is the same as that of COREG, i.e., $D_1 = Euclidean$, $D_2 = Mahalanobis$, and $k_1 = k_2 = 3$. Note that SELF2 is almost the same as COREG except that each regressor labels the unlabeled examples for itself instead of its peer regressor.

The maximum number of iterations T and the pool size s are set to 100 for ARTRE, SELF1 and SELF2, just as the same as that used for COREG.

Additionally, a kNN regressor using only the labeled data is evaluated as the baseline for the comparison, which is denoted by LABELED. The k value is set to 3. Note that the reported result of LABELED is the best result of such a kNN regressor which uses either Euclidean distance or

Mahalanobis distance.

Table VI reports on the results under different label rates, where the MSE of LABELED are tabulated in the third column, while the ratios of the final MSE of the other algorithms over the MSE of LABELED are tabulated in the fourth to the seventh columns. The lowest MSE ratio, i.e. the best performance among the compared semi-supervised algorithms, under each label rate has been boldfaced. Pairwise t-tests with significance level 0.05 are executed. Due to the page length, here we only present the p-values of comparing other methods against LABELED, as shown in Table VII, where the table entries with significant improvements are boldfaced.

First, let's look into the comparison between COREG, ARTRE and SELF1. Table VI shows that COREG is better than LABELED on all the data sets except on *Plane* under 10% label rate. By contrast, ARTRE is worse than LABELED on four data sets under every label rate, and SELF1 is worse than LABELED on five data sets under every label rate. Moreover, COREG achieves the lowest average MSE ratio under all label rates, while both the average MSE ratios of ARTRE and SELF1 are much higher than that of COREG. Pairwise t-tests with 0.05 significance level reveal that the final regression estimates of COREG are always significantly better than that of SELF1, almost always significantly better than that of LABELED except on *Plane* under 10% label rate, and almost always significantly better than that of ARTRE except that on *Friedman #1* ARTRE is better under 30% and 50% label rates and there is no significant difference under 10% label rate. It is evident that COREG is significantly better than both ARTRE and SELF1.

Second, let's look into the comparison between COREG and SELF2. Table VI shows that COREG is only worse than LABELED on *Plane* under 10% label rate, but SELF2 is worse than LABELED on *Plane* under all label rates. Moreover, on the 30 times (10 data sets × 3 label rates) of comparisons, COREG achieves the lowest MSE ratio for 20 times, while SELF2 only achieves the lowest MSE ratio for 9 times. These observations tell that COREG works better than SELF2. However, the average MSE ratio of SELF2 is close to that of COREG, and pairwise t-tests with 0.05 significance level disclose that COREG only significantly outperforms SEFL2 on 7 times of comparisons among the total 30 times of comparisons, i.e., on *Plane* under 10% label rate, on *2-d Mexican Hat*, *Plane* and *SinC* under 30% label rate, and on *2-d Mexican Hat*, *Gabor* and *Plane* under 50% label rate. So, the superiority of COREG to SELF2 is not so large as to ARTRE and SELF1. Remind that the only difference between SELF2 and COREG is that in COREG the two kNN regressors label examples for each other while in SELF2 the regressors label examples

TABLE VI

The MSE of Labeled and the ratios of other algorithms over Labeled on synthetic data sets

Label Rate	Data Sets	Labeled	Coreg	Artre	Self1	Self2
10%	*2d Mexican Hat*	2.98E-05	**0.653**	0.739	0.844	0.661
	3d Mexican Hat	3.56E-03	0.878	0.918	0.943	**0.868**
	Friedman #1	4.65E-03	0.958	**0.954**	1.036	0.959
	Friedman #2	2.43E-03	**0.915**	2.658	1.031	0.916
	Friedman #3	1.27E-02	0.968	1.349	1.015	**0.963**
	Gabor	1.69E-03	**0.816**	0.944	0.949	0.827
	Multi	2.89E-03	0.931	1.091	1.032	**0.928**
	Plane	9.41E-02	**1.010**	1.065	1.079	1.037
	Polynomial	2.29E-05	0.607	0.662	0.749	**0.596**
	SinC	2.09E-05	**0.551**	0.707	0.715	0.555
	avg.	N/A	**0.829**	1.109	0.939	0.831
30%	*2d Mexican Hat*	3.26E-06	**0.801**	0.889	0.922	0.812
	3d Mexican Hat	1.09E-03	**0.883**	0.968	0.994	0.889
	Friedman #1	2.85E-03	0.966	**0.910**	1.013	0.965
	Friedman #2	1.29E-03	**0.942**	2.231	1.017	0.944
	Friedman #3	8.36E-03	0.978	1.379	1.016	**0.976**
	Gabor	4.83E-04	**0.845**	0.971	0.980	0.852
	Multi	1.64E-03	0.948	1.031	1.019	**0.947**
	Plane	9.61E-02	**0.997**	1.027	1.032	1.007
	Polynomial	2.37E-06	**0.728**	0.794	0.849	0.732
	SinC	2.26E-06	**0.737**	0.840	0.886	0.747
	avg.	N/A	**0.882**	1.104	0.973	0.887
50%	*2d Mexican Hat*	1.20E-06	**0.860**	0.932	0.943	0.866
	3d Mexican Hat	6.39E-04	**0.902**	0.973	0.993	0.907
	Friedman #1	2.25E-03	0.975	**0.893**	1.008	0.962
	Friedman #2	9.60E-04	0.957	2.060	1.009	**0.956**
	Friedman #3	6.86E-03	**0.980**	1.383	1.010	**0.980**
	Gabor	2.76E-04	**0.882**	0.975	0.994	0.890
	Multi	1.28E-03	**0.957**	1.014	1.013	**0.957**
	Plane	9.41E-02	**0.996**	1.023	1.017	1.000
	Polynomial	7.66E-07	**0.812**	0.875	0.929	0.814
	SinC	8.18E-07	**0.802**	0.891	0.919	0.808
	avg.	N/A	**0.912**	1.102	0.983	0.914

TABLE VII

The p-Values of pairwise t-tests between the compared methods and Labeled on synthetic data sets

Label Rate	Data Set	Coreg	Artre	Self1	Self2
10%	*2d Mexican Hat*	**8.87E-35**	**2.08E-26**	**5.43E-09**	**1.92E-33**
	3d Mexican Hat	**2.31E-05**	**2.21E-03**	**4.63E-02**	**1.14E-05**
	Friedman #1	**5.50E-08**	**5.50E-10**	1.02E-05	**1.57E-07**
	Friedman #2	**6.14E-14**	7.51E-82	1.88E-03	**4.18E-14**
	Friedman #3	**3.48E-02**	6.16E-47	3.31E-01	**1.79E-02**
	Gabor	**1.27E-04**	1.88E-01	2.79E-01	**5.62E-04**
	Multi	**4.03E-09**	3.68E-12	2.19E-03	**1.74E-10**
	Plane	2.63E-01	1.57E-09	1.18E-11	4.59E-04
	Polynomial	**6.65E-21**	**3.33E-20**	**1.44E-09**	**6.32E-22**
	SinC	**7.29E-42**	**8.31E-22**	**8.35E-21**	**1.75E-41**
30%	*2d Mexican Hat*	**2.14E-28**	**1.28E-10**	**9.85E-07**	**5.27E-26**
	3d Mexican Hat	**2.24E-08**	1.32E-01	8.06E-01	**1.19E-07**
	Friedman #1	**2.01E-09**	**2.67E-31**	1.63E-02	**2.17E-09**
	Friedman #2	**7.98E-11**	3.49E-87	2.82E-02	**8.50E-11**
	Friedman #3	1.45E-01	2.69E-46	2.94E-01	1.13E-01
	Gabor	**6.51E-14**	1.29E-01	3.38E-01	**1.32E-12**
	Multi	**6.28E-10**	6.14E-05	1.30E-02	**3.54E-10**
	Plane	6.56E-01	1.10E-03	9.47E-05	3.72E-01
	Polynomial	**8.63E-21**	**1.07E-13**	**1.55E-08**	**7.93E-21**
	SinC	**1.11E-28**	**1.76E-15**	**7.75E-08**	**7.00E-28**
50%	*2d Mexican Hat*	**6.73E-21**	**2.94E-07**	**6.70E-06**	**6.70E-20**
	3d Mexican Hat	**2.07E-07**	1.18E-01	7.37E-01	**3.23E-07**
	Friedman #1	**7.02E-06**	**2.25E-37**	1.56E-01	**2.27E-06**
	Friedman #2	**1.72E-11**	5.07E-83	1.82E-01	**1.62E-11**
	Friedman #3	1.76E-01	6.31E-47	4.64E-01	1.68E-01
	Gabor	**2.83E-12**	1.06E-01	7.38E-01	**8.39E-10**
	Multi	**1.08E-08**	4.71E-02	8.11E-02	**6.78E-09**
	Plane	5.78E-01	2.01E-03	1.77E-02	9.74E-01
	Polynomial	**1.61E-14**	**7.19E-09**	**5.59E-03**	**7.85E-14**
	SinC	**5.10E-23**	**4.61E-09**	**1.52E-05**	**6.42E-22**

for themselves. Thus, the above results suggest that although letting the two regressors to label examples for each other is helpful, using two kNN regressors with different distance metrics and using Eq. 1 to select unlabeled examples contribute more to the performance of Coreg.

Note that with the increasing of the label rate, the reduction of MSE endowed by exploiting unlabeled examples seems decreasing. This is not strange since it can be perceived from the performance of LABELED that the initial regressors become stronger when more labeled training examples are available, and therefore are more difficult to be improved.

C. Experiments on Using kNN Regressors on Real-World Data Sets

Further, we use the real-world data sets shown in Table III to evaluate COREG and the other three semi-supervised learning algorithms described in Section V-B.2, i.e. ARTRE, SELF1 and SELF2. LABELED is still used as the baseline for comparison. All the experimental settings are as the same as that used in Section V-B.2.

Table VIII reports on the results under different label rates, where the MSE of LABELED are tabulated in the third column, while the ratios of the final MSE of the other algorithms over the MSE of LABELED are shown in the fourth to the seventh columns. The lowest MSE ratio under each label rate, i.e. the best performance of the compared semi-supervised algorithms, has been boldfaced. Pairwise t-tests with significance level 0.05 are executed. Due to the page length, here we only present the p-values of comparing other methods against LABELED, as shown in Table IX, where the table entries with significant improvements are boldfaced.

First, let's look into the comparison between COREG, ARTRE and SELF1. Table VIII shows that COREG always achieves better performance than LABELED. By contrast, ARTRE is always worse than LABELED on *kin8nm_2000* and SELF1 is always worse than LABELED. Moreover, the average MSE ratio of COREG is smaller than 1.0 under all label rates, while both the average MSE ratios of ARTRE and SELF1 are bigger than 1.0.

Second, let's look into the comparison between COREG and SELF2. Table VIII shows that COREG is always better than LABELED but SELF2 is worse than LABELED on *no2* under 10% label rate. The average MSE ratio of COREG is smaller than that of SELF2. Moreover, on the 15 times (5 data sets × 3 label rates) of comparisons, COREG achieves the lowest MSE ratio for 8 times, while SELF2 only achieves the lowest MSE ratio for 2 times.

Pairwise t-tests with significance level 0.05 indicate that COREG always significantly outperforms LABELED and SELF1; it is significantly better than ARTRE on *kin8nm_2000* and *pollen* under all label rates and on *no* under 10% and 30% label rates; it is significantly better than

TABLE VIII

The MSE of Labeled and the ratios of other algorithms over Labeled on real-world data sets

Label Rate	Data Set	Labeled	Coreg	Artre	Self1	Self2
10%	*chscase.census6*	2.97E-02	0.950	**0.949**	1.041	0.974
	kin8nm_2000	2.08E-02	**0.963**	1.213	1.028	0.964
	no2	1.45E-01	**0.950**	0.982	1.109	1.017
	pollen	1.11E-01	**0.912**	0.954	1.008	0.924
	puma8NH_2000	4.04E-02	0.966	**0.910**	1.021	0.965
	avg.	N/A	**0.976**	1.010	1.032	0.995
30%	*chscase.census6*	2.92E-02	0.948	**0.926**	1.039	0.960
	kin8nm_2000	1.59E-02	**0.966**	1.299	1.018	**0.966**
	no2	1.37E-01	**0.936**	0.948	1.027	0.951
	pollen	1.10E-01	**0.914**	0.949	1.012	0.918
	puma8NH_2000	3.56E-02	0.970	**0.877**	1.009	0.972
	avg.	N/A	**0.968**	1.008	1.016	0.975
50%	*chscase.census6*	2.84E-02	0.950	**0.930**	1.031	0.956
	kin8nm_2000	1.41E-02	0.978	1.327	1.017	**0.974**
	no2	1.35E-01	**0.930**	0.933	1.002	0.936
	pollen	1.10E-01	**0.911**	0.944	1.008	0.914
	puma8NH_2000	3.34E-02	0.974	**0.869**	1.004	0.975
	avg.	N/A	**0.963**	1.009	1.008	0.964

Self2 on *chscase.census6* under all label rates and on *no2* and *pollen* under 10% and 30% label rates. The above observations tell that Coreg is superior to the compared algorithms.

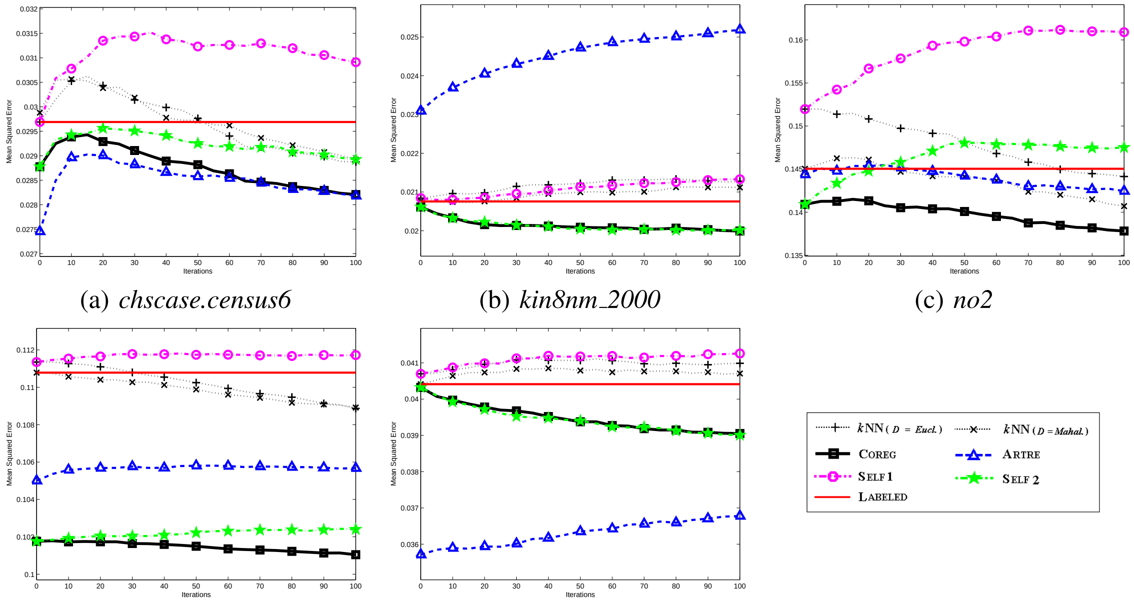
Since experiments on real-world data sets may exhibit the performance of the compared algorithms on real-world tasks better than experiments on synthetic data sets, we study further the MSE of different algorithms at different iterations, as shown in Figs. 1 to 3. In these figures, besides the compared algorithms, the MSE of the two kNN regressors used in Coreg are also depicted. Note that in each figure, every curve contains 101 points corresponding to the 100 learning iterations in addition to the initial condition, where only 11 of them are explicitly depicted for better presentation.

Figs. 1 to 3 show that the MSE of Coreg usually keeps on decreasing as the learning process proceeds, which suggests that Coreg improves its regression estimates by exploiting unlabeled examples. Self1 is obviously incompetent. After using unlabeled data, the perfor-

TABLE IX

THE p-VALUES OF PAIRWISE t-TESTS BETWEEN THE COMPARED METHODS AND LABELED ON REAL-WORLD DATA SETS

Label Rate	Data Set	COREG	ARTRE	SELF1	SELF2
10%	*chscase_census6*	**9.86E-04**	**1.44E-03**	2.04E-02	8.73E-02
	kin8nm_2000	**4.05E-05**	3.39E-40	2.86E-03	**6.83E-05**
	no2	**1.21E-06**	9.15E-02	1.16E-13	1.06E-01
	pollen	**2.37E-42**	**4.31E-21**	8.23E-02	**4.79E-37**
	puma8NH_2000	**5.19E-06**	**9.64E-24**	7.00E-03	**2.26E-06**
30%	*chscase_census6*	**2.35E-04**	**2.44E-07**	8.45E-03	**5.21E-03**
	kin8nm_2000	**1.05E-05**	2.62E-51	1.71E-02	**1.33E-05**
	no2	**6.10E-12**	**9.54E-09**	7.44E-03	**1.01E-07**
	pollen	**6.83E-46**	**2.24E-27**	1.81E-03	**2.38E-44**
	puma8NH_2000	**2.06E-05**	**4.43E-39**	2.33E-01	**7.87E-05**
50%	*chscase_census6*	**3.02E-04**	**6.60E-08**	2.72E-02	**1.03E-03**
	kin8nm_2000	**3.48E-03**	4.28E-58	3.23E-02	**4.29E-04**
	no2	**1.59E-12**	**3.29E-13**	8.12E-01	**2.69E-10**
	pollen	**3.01E-50**	**2.53E-28**	2.76E-02	**1.30E-48**
	puma8NH_2000	**1.18E-04**	**1.48E-41**	5.07E-01	**2.88E-04**



(a) *chscase.census6*

(b) *kin8nm_2000*

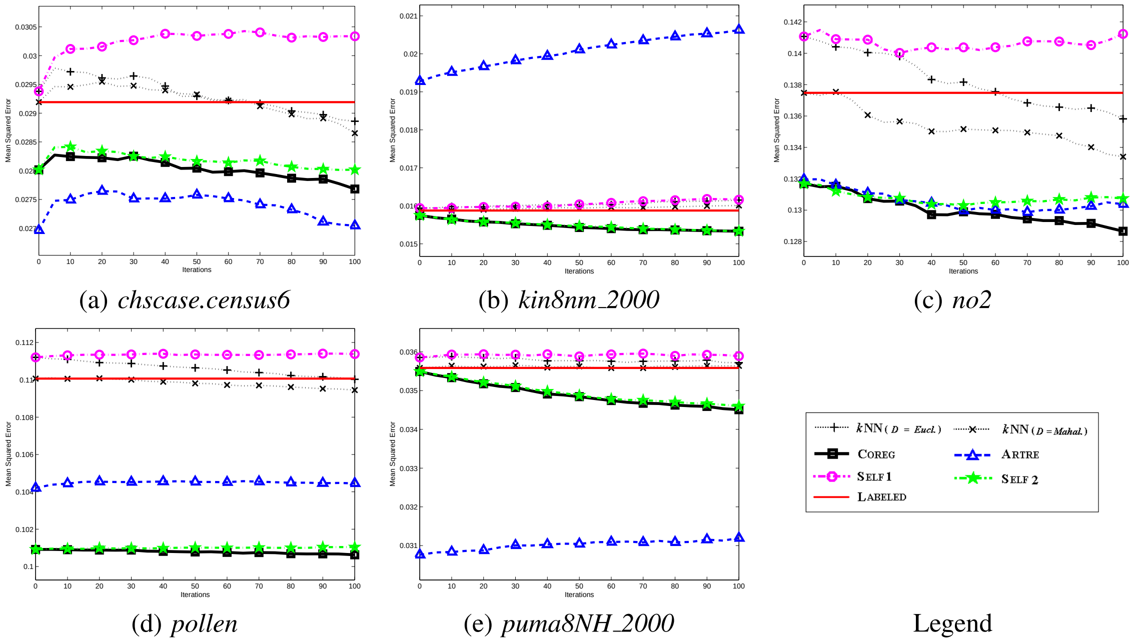
(c) *no2*

(d) *pollen*

(e) *puma8NH_2000*

Legend

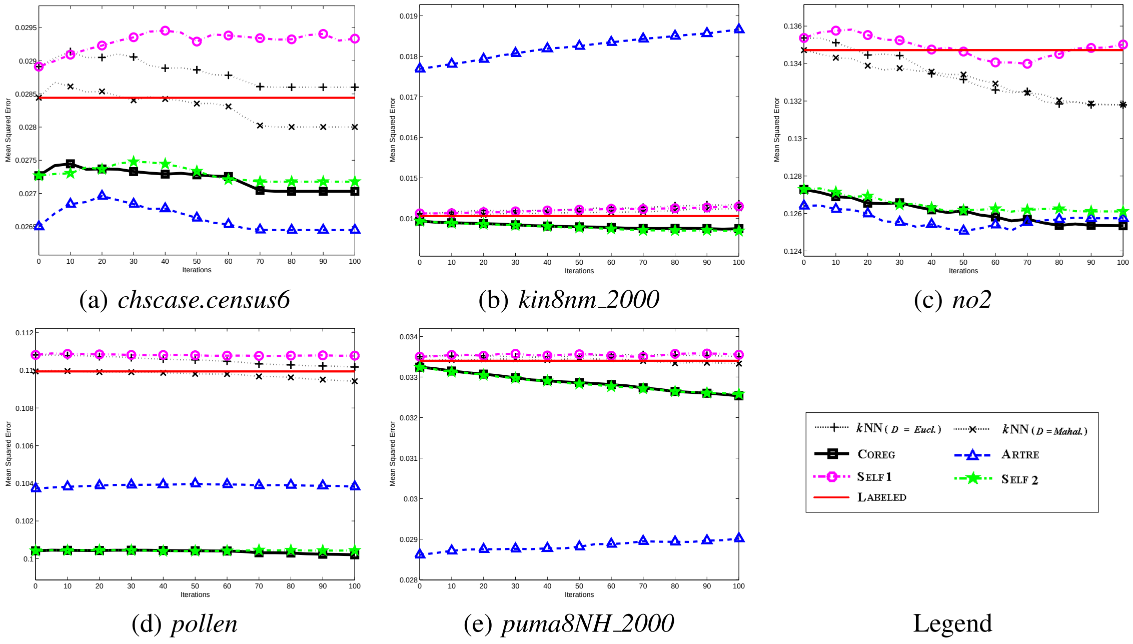
Fig. 1. Comparison on MSE at different iterations when the label rate is 10%



(a) *chscase.census6* (b) *kin8nm_2000* (c) *no2*

(d) *pollen* (e) *puma8NH_2000* Legend

Fig. 2. Comparison on MSE at different iterations when the label rate is 30%



(a) *chscase.census6* (b) *kin8nm_2000* (c) *no2*

(d) *pollen* (e) *puma8NH_2000* Legend

Fig. 3. Comparison on MSE at different iterations when the label rate is 50%

mance of ARTRE is often degraded. In fact, the low MSE ratios of ARTRE on four data sets in Table VIII owe to that its initial regression estimates on these data sets are better than that of LABELED. For example, ARTRE achieves the lowest MSE ratio among the compared algorithms on *chscase.census6* and *puma8NH_2000*, but its performance actually degenerates on *puma8NH_2000* under all label rates and on *chscase.census6* under 10% and 30% label rates after it exploits the unlabeled data. Similarly, although the final MSE of SELF2 is almost always better than LABELED, the performance of SELF2 actually degenerates on *chscase.census6*, *no2* and *pollen* under 10% label rate, on *chscase.census6* and *pollen* under 30% label rate, and on *pollen* under 50% label rate. The above observations confirm that the performance of COREG is superior to the compared algorithms.

D. Experiments on Using Other Regressors

As mentioned before, COREG can be easily used with other kinds of regressors. After employing two kNN regressors to select and label the unlabeled examples, the predictions can be made by other kinds of regressors trained from the two augmented labeled training sets instead of the kNN regressors.

We run experiments on the ten synthetic data sets under 10% label rate, and the results are shown in Table X. We have tried two widely used regressors, i.e. linear regressor and support vector regressor. Besides COREG, the compared algorithms in Section V-B.2 are also evaluated. Here the parameter settings are as the same as that used in Section V-B.2. Pairwise t-tests with significance level 0.05 are executed. Due to the page length, here we only present the p-values of comparing other methods against LABELED, as shown in Table XI, where the table entries with significant improvements are boldfaced.

It can be observed from Table X that COREG always achieve better regression estimates than LABELED except on *Polynomial* when linear regressors are used. COREG almost always achieves the lowest MSE ratio expect on *Gabor* and *Polynomial* when linear regressors are used, and on *Friedman #1,#2* when support vector regressor are used. Moreover, the average MSE error ratio of COREG is lower than that of ARTRE, SELF1 and SELF2. These observations tell that COREG is superior to the compared algorithms no matter whether linear regressors or support vector regressors are used.

However, Table X shows that when linear regressors are used, the improvements from exploiting unlabeled data are not significant except for COREG on *Plane*; while when support vector

TABLE X

THE MSE OF LABELED AND THE RATIOS OF OTHER ALGORITHMS OVER LABELED UNDER 10% LABEL RATE

Data set	Linear Regressor					Support Vector Regressor				
	LABELED	COREG	ARTRE	SELF1	SELF2	LABELED	COREG	ARTRE	SELF1	SELF2
2d Mexican Hat	1.23E-01	**0.999**	**0.999**	1.000	**0.999**	1.52E-01	**0.959**	1.005	0.976	0.985
3d Mexican Hat	1.33E-02	**0.998**	1.000	1.000	0.999	1.36E-02	**0.993**	0.997	0.996	0.995
Friedman #1	8.35E-03	**0.995**	0.998	0.999	0.998	8.60E-03	0.996	**0.985**	0.991	0.988
Friedman #2	6.27E-03	**0.998**	2.105	0.999	**0.998**	6.31E-03	0.998	1.103	0.998	**0.996**
Friedman #3	1.85E-02	**0.998**	1.270	1.000	0.999	2.48E-02	**0.933**	0.913	0.971	0.966
Gabor	2.77E-02	0.997	**0.995**	0.997	0.996	3.04E-02	**0.969**	0.990	0.991	0.984
Multi	3.38E-03	**0.993**	1.016	1.005	1.004	3.41E-03	**0.988**	1.033	1.015	1.013
Plane	7.34E-02	**0.977**	1.018	1.016	1.014	7.80E-02	**0.937**	0.984	0.982	0.981
Polynomial	8.82E-03	1.011	**0.997**	1.003	0.999	1.06E-02	**0.946**	0.981	0.965	0.958
SinC	2.25E-02	**0.996**	0.999	0.998	0.997	2.66E-02	**0.977**	1.001	0.982	0.978
avg.	N/A	**0.996**	1.140	1.002	1.000	N/A	**0.969**	0.999	0.987	0.985

TABLE XI

THE p-VALUES OF PAIRWISE t-TESTS BETWEEN THE COMPARED METHODS AND LABELED ON SYNTHETIC DATA SETS UNDER 10% LABEL RATE

Data set	Linear Regressor				Support Vector Regressor			
	COREG	ARTRE	SELF1	SELF2	COREG	ARTRE	SELF1	SELF2
2d Mexican Hat	3.72E-01	3.57E-01	2.47E-01	4.12E-01	**1.75E-13**	6.57E-01	**3.76E-05**	**2.27E-03**
3d Mexican Hat	8.50E-01	9.73E-01	9.98E-01	9.41E-01	5.39E-01	7.72E-01	7.62E-01	6.92E-01
Friedman #1	3.17E-01	7.21E-01	8.63E-01	7.47E-01	5.09E-01	**1.41E-02**	1.19E-01	5.66E-02
Friedman #2	5.50E-01	1.33E-104	7.91E-01	5.32E-01	6.67E-01	8.13E-28	6.52E-01	3.55E-01
Friedman #3	6.81E-01	1.56E-48	9.60E-01	8.32E-01	**2.58E-09**	**5.00E-15**	**1.20E-02**	**3.25E-03**
Gabor	5.38E-01	3.20E-01	6.21E-01	4.19E-01	**2.62E-05**	2.38E-01	2.65E-01	**3.34E-02**
Multi	1.53E-01	2.51E-03	3.69E-01	4.72E-01	**8.36E-03**	1.80E-06	1.04E-02	1.84E-02
Plane	**5.59E-05**	1.53E-02	2.28E-02	4.52E-02	**5.67E-21**	5.38E-02	**2.13E-02**	**1.90E-02**
Polynomial	1.70E-02	6.29E-01	5.29E-01	9.12E-01	**7.94E-09**	1.47E-01	**1.60E-03**	**2.68E-05**
SinC	5.10E-01	8.18E-01	9.98E-01	6.25E-01	**1.73E-02**	9.90E-01	9.34E-02	**2.55E-02**

regressors are used, the improvements from exploiting unlabeled data are not significant on *3d Mexican Hat*, *Friedman #1* and *Friedman #2* except for ARTRE on *Friedman #1*. We have also conducted experiments under 30% and 50% label rates, and found that although COREG can still benefit from exploiting the unlabeled examples, the improvements are not significant. The compared algorithms could not work well either.

In fact, if we compare Tables X and VI, we can find that when linear regressors or support vector regressors are used, the gains from exploiting unlabeled data are not as apparent as what have been achieved when kNN regressors are used. We think this owes to two reasons. First, linear regressors and support vector regressors exploit global information for regression estimates. When there are lots of labeled training examples, exploiting a limited number of unlabeled examples would not significantly change the global information, and therefore the regression estimates could not be apparently improved. By contrast, kNN regressors exploit local information, which could benefit much from the additional labeled training examples since in many local areas the examples become more dense. Second, remind that the key assumption in semi-supervised regression is the "manifold assumption", i.e. the local "smoothness" in the feature space. The extension to linear regressors and support vector regressors described in this section is quite naive since it still uses kNN regressors to select and label the unlabeled examples. It is likely that the "smoothness" induced by kNN regressors is somewhat different from what is suitable for linear regressors and support vector regressors. In other words, if we use linear regressors or support vector regressors themselves to accomplish the process of selecting and labelling the unlabeled examples, better performance might be achieved by exploiting unlabeled examples, which is a future extension of COREG.

E. Summary of Experimental Results

Overall, the experimental results reported in this section show that:

- The COREG algorithm can effectively exploit unlabeled examples to help improve regression estimates. In most cases its improvement is larger than the compared algorithms. In particular, when using fixed k value but different distance metrics, the improvement of COREG is always the biggest among all the compared algorithms. Moreover, the final regression estimates of COREG is usually the best.
- According to the improvements brought by exploiting unlabeled examples, the three configurations of COREG in descending order are: Using fixed k value but different distance metrics, using fixed distance metric but different k values, and using different k values and

different distance metrics. In order to exert the advantages of using different metrics and different k values simultaneously, a careful study on the cooperation of metrics and k values should be performed.

- The ARTRE algorithm which implements co-training through artificially deriving the second view from the original attributes often degrades the performance after exploiting unlabeled examples. The fact that COREG is usually superior to ARTRE suggests that exploiting different distance metrics and/or different k values is better than exploiting artificial view when co-training style algorithms are applied to regression problems which lack sufficient and redundant views.
- The SELF1 algorithm which implements single-learner self-training never achieves better regression estimates than COREG. This suggests that the co-training style used in COREG is better than the standard self-training style for exploiting the unlabeled data in regression problems.
- The SELF2 algorithm which combines two self-trained regressors for final prediction could be helpful on many regression data sets, although inferior to COREG. Note that the only difference between SELF2 and COREG is that in COREG the two kNN regressors label examples for each other while in SELF2 they label examples for themselves. This suggests that although letting the two regressors to label examples for each other is also helpful, the more important mechanisms in COREG are to use two diverse kNN regressors and to pick the unlabeled example which makes the regressor most consistent with the labeled example set to label.
- When the number of labeled training examples increases, the gains through exploiting unlabeled examples usually decrease since the regressors trained on the labeled training examples become stronger, which are more difficult to be improved.
- COREG can be extended to use other kinds of regressors. In order to get more gains from unlabeled examples, it may be better to involve the regressors in the prediction process as well as in the process of selecting and labelling unlabeled examples.

VI. CONCLUSION

Previous research on semi-supervised learning mainly focuses on semi-supervised classification. This paper extends our previous work [50] which describes one of the early efforts on semi-supervised regression. In particular, this paper proposes a co-training style semi-supervised regression algorithm COREG. This algorithm employs two different k-nearest neighbor regressors. In every learning iteration, each regressor labels the unlabeled example which can be most confidently labeled for the other regressor, where the labeling confidence is estimated through considering the consistency of the regressor with the labeled example set. The final prediction is made by averaging the predictions of both the refined kNN regressors. Analysis and experiments show that COREG can effectively exploit unlabeled data to improve the regression estimates.

This paper uses kNN regressor as the base learner, but the key idea of COREG, i.e. regarding the labeling of the unlabeled example which makes the regressor most consistent with the labeled example set as with the most confidence, can also be used with other base learners. A straightforward extension of COREG has been studied in this paper. In order to get more gains from unlabeled data, it may be better to involve the base learner in both the process of prediction and the process of selecting and labelling unlabeled examples. Designing semi-supervised regression algorithms along with this way is an interesting issue to be explored in the future.

Currently there are many semi-supervised classification algorithms. Studying the relationship between semi-supervised classification and semi-supervised regression, and developing other kinds of semi-supervised regression algorithms, are also interesting issues to be investigated in the future.

It has been reported that exploiting unlabeled examples is not always beneficial and sometimes the performance may be degenerated [28], [35], which has also been observed in our experiments. Although some explanations owing the deterioration to invalid model assumption [13], [28], [35] or inconsistent data distribution [39], at present there is no solid theory guiding the exploitation of unlabeled examples. Trying to establish such a theoretical framework is a great challenge of semi-supervised learning.

ACKNOWLEDGMENTS

We want to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] N. Abe and H. Mamitsuka, “Query learning strategies using boosting and bagging,” in *Proceedings of the 15th International Conference on Machine Learning*, Madison, WI, 1998, pp. 1–9.

[2] S. Abney, “Bootstrapping,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, 2002, pp. 360–367.

[3] M.-F. Balcan, A. Blum, and K. Yang, “Co-training and expansion: Towards bridging theory and practice,” in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 89–96.

[4] M. Belkin and P. Niyogi, “Semi-supervised learning on Riemannian manifolds,” *Machine Learning*, vol. 56, no. 1-3, pp. 209–239, 2004.

[5] M. Belkin, P. Niyogi, and V. Sindhwani, “On manifold regularization,” in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, Savannah Hotel, Barbados, 2005, pp. 17–24.

[6] C. Blake, E. Keogh, and C. J. Merz, “UCI repository of machine learning databases,” [http://www.ics.uci.edu/∼mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, CA, 1998.

[7] A. Blum and S. Chawla, “Learning from labeled and unlabeled data using graph mincuts,” in *Proceedings of the 18th International Conference on Machine Learning*, Williamston, MA, 2001, pp. 19–26.

[8] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proceedings of the 11th Annual Conference on Computational Learning Theory*, Madison, WI, 1998, pp. 92–100.

[9] U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel, “Efficient co-regularised least squares regression,” in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006, pp. 137–144.

[10] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

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

[12] M. Collins and Y. Singer, “Unsupervised models for named entity classifications,” in *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, College Park, MD, 1999, pp. 100–110.

[13] F. G. Cozman and I. Cohen, “Unlabeled data can degrade classification performance of generative classifiers,” in *Proceedings of the 15th International Conference of the Florida Artificial Intelligence Research Society*, Pensacola, FL, 2002, pp. 327–331.

[14] B. V. Dasarathy, *Nearest Neighbor Norms: NN Pattern Classification Techniques*. Los Alamitos, CA: IEEE Computer Society Press, 1991.

[15] S. Dasgupta, M. Littman, and D. McAllester, “PAC generalization bounds for co-training,” in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, pp. 375–382.

[16] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[17] T. G. Dietterich, “Ensemble methods in machine learning,” in *Lecture Notes in Computer Science 1867*, J. Kittler and F. Roli, Eds. Berlin: Springer, 2000, pp. 1–15.

[18] M. O. Franz, Y. Kwon, C. E. Rasmussen, and B. Schökopf, “Semi-supervised kernel regression using whitened function classes,” in *Lecture Notes in Computer Science 3175*, C. E. Rasmussen, H. H. Bülthoff, B. Schölkopf, and M. A. Giese, Eds. Berlin: Springer, 2004, pp. 18–26.

[19] A. Fujino, N. Ueda, and K. Saito, “A hybrid generative/discriminative approach to semi-supervised classifier design,” in *Proceedings of the 20th National Conference on Artificial Intelligence*, Pittsburgh, PA, 2005, pp. 764–769.

[20] S. Goldman and Y. Zhou, "Enhancing supervised learning with unlabeled data," in *Proceedings of the 17th International Conference on Machine Learning*, San Francisco, CA, 2000, pp. 327–334.

[21] J. V. Hansen, "Combining predictors: Meta machine learning methods and bias/variance and ambiguity decompositions," Ph.D. dissertation, Department of Computer Science, University of Aarhus.

[22] R. Hwa, M. Osborne, A. Sarkar, and M. Steedman, "Corrected co-training for statistical parsers," in *Working Notes of the ICML'03 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, Washington, DC, 2003.

[23] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proceedings of the 16th International Conference on Machine Learning*, Bled, Slovenia, 1999, pp. 200–209.

[24] M. Li and Z.-H. Zhou, "Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples," *IEEE Transactions on Systems, Man and Cybernetics - Part A*, vol. 38, 2008.

[25] R. P. Lippmann, "Pattern classification using neural networks," *IEEE Communications*, vol. 27, no. 11, pp. 47–64, 1989.

[26] D. J. Miller and H. S. Uyar, "A mixture of experts classifier with learning based on both labelled and unlabelled data," in *Advances in Neural Information Processing Systems 9*, M. Mozer, M. I. Jordan, and T. Petsche, Eds. Cambridge, MA: MIT Press, 1997, pp. 571–577.

[27] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proceedings of the 9th ACM International Conference on Information and Knowledge Management*, Washington, DC, 2000, pp. 86–93.

[28] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Machine Learning*, vol. 39, no. 2-3, pp. 103–134, 2000.

[29] D. Pierce and C. Cardie, "Limitations of co-training for natural language learning from large data sets," in *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, Pittsburgh, PA, 2001, pp. 1–9.

[30] A. Pozdnoukhov and S. Bengio, "Semi-supervised kernel methods for regression estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5, Toulouse, France, 2006, pp. 577–580.

[31] G. Ridgeway, D. Madigan, and T. Richardson, "Boosting methodology for regression problems," in *Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, 1999, pp. 152–161.

[32] E. Riloff and R. Jones, "Learning dictionaries for information extraction by multi-level bootstrapping," in *Proceedings of the 16th National Conference on Artificial Intelligence*, Orlando, FL, 1999, pp. 474–479.

[33] A. Sarkar, "Applying co-training methods to statistical parsing," in *Proceedings of the 2nd Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, PA, 2001, pp. 95–102.

[34] H. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the 5th ACM Workshop on Computational Learning Theory*, Pittsburgh, PA, 1992, pp. 287–294.

[35] B. Shahshahani and D. Landgrebe, "The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 5, pp. 1087–1095, 1994.

[36] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: From transductive to semi-supervised learning," in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005, pp. 824–831.

[37] ——, "A co-regularized approach to semi-supervised learning with multiple views," in *Working Notes of the ICML'05 Workshop on Learning with Multiple Views*, Bonn, Germany, 2005.

[38] M. Steedman, M. Osborne, A. Sarkar, S. Clark, R. Hwa, J. Hockenmaier, P. Ruhlen, S. Baker, and J. Crim, "Bootstrapping

statistical parsers from small data sets," in *Proceedings of the 11th Conference on the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary, 2003, pp. 331–338.

[39] Q. Tian, J. Yu, Q. Xue, and N. Sebe, "A new analysis of the value of unlabeled data in semi-supervised learning for image retrieval," in *Proceedings of the IEEE International Conference on Multimedia Expo*, Taibei, 2004, pp. 1019–1022.

[40] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[41] P. Vlachos, "StatLib project repository," [http://lib.stat.cmu.edu], Department of Statistics, Carnegie Mellon University, Pittsburgh, PA, 2000.

[42] M. Wang, X.-S. Hua, Y. Song, L.-R. Dai, and H.-J. Zhang, "Semi-supervised kernel regression," in *Proceedings of the 6th IEEE International Conference on Data Mining*, Hong Kong, China, 2006, pp. 1130–1135.

[43] W. Wang and Z.-H. Zhou, "Analyzing co-training style algorithms," in *Proceedings of the 18th European Conference on Machine Learning*, Warsaw, Poland, 2007.

[44] J. A. E. Weston, M. O. Stitson, A. Gammerman, V. Vovk, and V. Vapnik, "Experiments with support vector machines," Royal Holloway University of London, London, UK, Tech. Rep. CSD-TR-96-19, 1996.

[45] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Cambridge, MA, 1995, pp. 189–196.

[46] D. Zhou, B. Schölkopf, and T. Hofmann, "Semi-supervised learning on directed graphs," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 1633–1640.

[47] Y. Zhou and S. Goldman, "Democratic co-learning," in *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, Boca Raton, FL, 2004, pp. 594–602.

[48] Z.-H. Zhou, K.-J. Chen, and H.-B. Dai, "Enhancing relevance feedback in image retrieval using unlabeled data," *ACM Transactions on Information Systems*, vol. 24, no. 2, pp. 219–244, 2006.

[49] Z.-H. Zhou, K.-J. Chen, and Y. Jiang, "Exploiting unlabeled data in content-based image retrieval," in *Proceedings of the 15th European Conference on Machine Learning*, Pisa, Italy, 2004, pp. 525–536.

[50] Z.-H. Zhou and M. Li, "Semi-supervised learning with co-training," in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, 2005, pp. 908–913.

[51] ——, "Tri-training: exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.

[52] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artificial Intelligence*, vol. 137, no. 1-2, pp. 239–263, 2002.

[53] Z.-H. Zhou, D.-C. Zhan, and Q. Yang, "Semi-supervised learning with very few labeled training examples," in *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, Vancouver, Canada, 2007, pp. 675–680.

[54] X. Zhu, "Semi-supervised learning literature survey," Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, Tech. Rep. 1530, 2006, http://www.cs.wisc.edu/∼jerryzhu/pub/ssl_survey.pdf.

[55] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proceedings of the 20th International Conference on Machine Learning*, Washington, DC, 2003, pp. 912–919.

Zhi-Hua Zhou (S'00-M'01-SM'06) received the BSc, MSc and PhD degrees in computer science from Nanjing University, China, in 1996, 1998 and 2000, respectively, all with the highest honors. He joined the Department of Computer Science & Technology at Nanjing University as a lecturer in 2001, and is Cheung Kong professor and head of the LAMDA group at present. His research interests are in artificial intelligence, machine learning, data mining, information retrieval, pattern recognition, evolutionary computation and neural computation. In these areas he has published over 50 papers in leading international journals or conference proceedings. He has won various awards/honors including the National Science & Technology Award for Young Scholars of China (2006), the Award of National Science Fund for Distinguished Young Scholars of China (2003), the National Excellent Doctoral Dissertation Award of China (2003), the Microsoft Young Professorship Award (2006), etc. He is on the editorial boards of *Knowledge and Information Systems*, *Artificial Intelligence in Medicine*, *International Journal of Data Warehousing and Mining*, *Journal of Computer Science & Technology*, *Frontiers of Computer Science in China*, *International Journal of Software and Informatics*, *Journal of Software* and *Journal of Frontiers of Computer Science and Technology*, and was guest editor/co-editor of journals including *ACM/Springer Multimedia Systems* and *The Computer Journal*. He served as the program committee chair of PAKDD'07, vice chair of ICDM'06, PRICAI'06, etc., program committee member for various international conferences including ICML, ECML, SIGKDD, ICDM, and chaired a number of native conferences. He is a senior member of China Computer Federation (CCF) and the vice chair of the CCF Artificial Intelligence & Pattern Recognition Society, an executive committee member of Chinese Association of Artificial Intelligence (CAAI) and the chair of the CAAI Machine Learning Society, a member of AAAI and ACM, and a senior member of IEEE and IEEE Computer Society.

Ming Li received the BSc degree in computer science from Nanjing University, China, in 2003. Currently, he is a PhD candidate at the Department of Computer Science & Technology of Nanjing University, and is a member of the LAMDA Group. His research interests mainly include machine learning and data mining, especially in learning with labeled and unlabeled examples. He has won a number of awards including the Microsoft Fellowship Award (2005), the HP Chinese Excellent Student Scholarship (2005), the Outstanding Graduate Student of Nanjing University (2006), etc. He won the PAKDD'06 Data Mining Competition Open Category Champion with other LAMDA members.