Chapter 7
Semi-supervised Learning

Mohamed Farouk Abdel Hady and Friedhelm Schwenker

Abstract. In traditional supervised learning, one uses ”labeled” data to build a model. However, labeling the training data for real-world applications is difficult, expensive, or time consuming, as it requires the effort of human annotators sometimes with specific domain experience and training. There are implicit costs associated with obtaining these labels from domain experts, such as limited time and financial resources. This is especially true for applications that involve learning with large number of class labels and sometimes with similarities among them. Semi-supervised learning (SSL) addresses this inherent bottleneck by allowing the model to integrate part or all of the available unlabeled data in its supervised learning. The goal is to maximize the learning performance of the model through such newly-labeled examples while minimizing the work required of human annotators. Exploiting unlabeled data to help improve the learning performance has become a hot topic during the last decade and it is divided into four main directions: SSL with graphs, SSL with generative models, semi-supervised support vector machines and SSL by disagreement (SSL with committees). This survey article provides an overview to research advances in this branch of machine learning.

1 Introduction

Supervised learning algorithms require a large amount of labeled training data in order to construct models with high prediction performance, see Figure 1. In many practical data mining applications such as computer-aided medical diagnosis [38], remote sensing image classification [49], speech recognition [32], email classification [33], or automated classification of text documents [44, 45], there is often an extremely inexpensive large pool of unlabeled data available. However, the data

Mohamed Farouk Abdel Hady · Friedhelm Schwenker
Institute of Neural Information Processing
University of Ulm
D-89069 Ulm, Germany
e-mail: {mohamed.abdel-hady,friedhelm.schwenker}@uni-ulm.de

M. Bianchini et al. (Eds.): *Handbook on Neural Information Processing*, ISRL 49, pp. 215–239.
DOI: 10.1007/978-3-642-36657-4_7

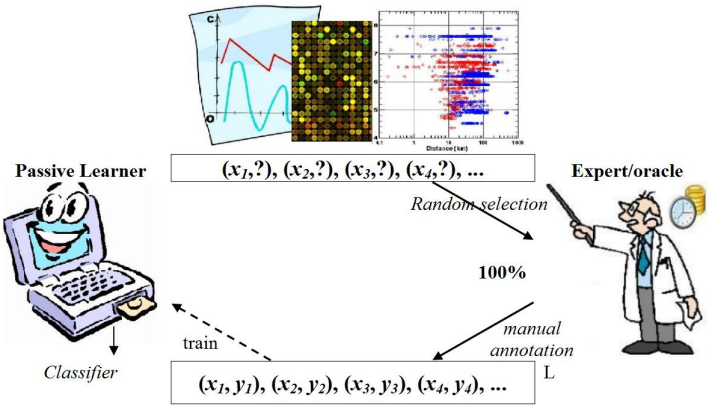


Fig. 1 Graphical illustration of traditional supervised learning

labeling process is often difficult, tedious, expensive, or time consuming, as it requires the efforts of human experts or special devices. Due to the difficulties in incorporating unlabeled data directly into traditional supervised learning algorithms such as support vector machines and RBF neural networks and the lack of a clear understanding of the value of unlabeled data in the learning process, the study of semi-supervised learning attracted attention only after the middle of 1990s. As the demand for automatic exploitation of unlabeled data increases, semi-supervised learning has become a hot topic.

In computer-aided diagnosis (CAD), mammography is a specific type of imaging that uses a low-dose x-ray system to examine breasts and it is used to aid in the early detection and diagnosis of breast diseases in women. There is a large number of mammographic images that can be obtained from routine examination but it is difficult to ask a physician or radiologist to search all images and highlight the abnormal areas of calcification that may indicate the presence of cancer. If we use supervised learning techniques to build a computer software to highlight these areas on the images, based on limited amount of diagnosed training images, it may be difficult to get an accurate diagnosis software. Then a question arises: can we exploit the abundant undiagnosed images [38] with the few diagnosed images to construct a more accurate software.

For remote sensing applications, the remote sensing sensors can produce data in large number of spectral bands. The objective of using such high resolution sensors is to discriminate among more ground cover classes and hence obtain a better understanding about the nature of the materials that cover the surface of the Earth. This large number of classes and large number of spectral bands require a large number of labeled training examples (pixels) from all the classes of interest. The class labels of such training examples are usually very expensive and time consuming to acquire [49]. The reason is that identifying the ground truth of the data must be gathered by visual inspection of the scene near the same time that the data is

being taken, by using an experienced analyst based on their spectral responses, or by other means. In any case, usually only a limited number of training examples can be obtained. The purpose of SSL is to study how to reduce the small sample size problem by using unlabeled data that may be available in large number and with no extra cost. In the machine learning literature, there are mainly three paradigms for addressing the problem of combining labeled and unlabeled data to boost the performance: *semi-supervised learning*, *transductive learning* and *active learning*. *Semi-supervised learning* (*SSL*) refers to methods that attempt to take advantage of unlabeled data for supervised learning (*semi-supervised classification*), see Figure 2, or to incorporate prior information such as class labels, pairwise constraints or cluster membership in the context of unsupervised learning (*semi-supervised clustering*). *Transductive learning* [52] refers to methods which also attempt to exploit unlabeled examples but assuming that the unlabeled examples are exactly the test examples. That is, the test data set is known in advance and the goal of learning is to optimize the classification performance on the given test set. *Active learning* [48], sometimes called *selective sampling* refers to methods which assume that the given learning algorithm has control on the selection of the input training data such that it can select the most important examples from a pool of unlabeled examples, then an oracle such as a human expert is asked for labeling these examples, where the aim is to minimize data utilization. The most popular algorithms are *uncertainty sampling* (*US*) and *query by committee* (*QBC*) sampling. The former trains a single classifier and then query the unlabeled example on which the classifier is least confident [37]; the latter constructs multiple classifiers and then query the unlabeled example on which the classifiers disagree to the most [24]. In the remainder of this article is organized as follows: brief introduction to semi-supervised learning is given in the next section, and then the different semi-supervised learning techniques are introduced in the following sections. Section 8 presents the combination of semi-supervised learning with active learning. Finally, we conclude in section 9.

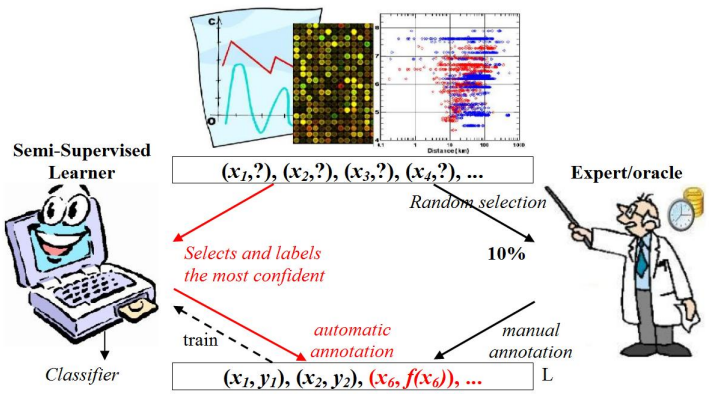


Fig. 2 Graphical illustration of semi-supervised learning

2 Semi-supervised Learning

Let $L = \{(x_i, y_i) | x_i \in \mathbb{R}^d, y_i \in \Omega, i = 1, \ldots, l\}$ be the set of labeled training examples where each example is described by a d-dimensional feature vector $x_i \in \mathbb{R}^d$, y_i denotes the class label of x_i and $\Omega = \{\omega_1, \ldots, \omega_K\}$ is the set of target classes (ground truth). Also let $U = \{x_j^* | j = 1, \ldots, u\}$ be the set of unlabeled data; usually $l << u$. The recent research on semi-supervised learning (*SSL*) concentrates into four directions: *semi-supervised classification* [13, 45, 33, 57, 61, 38], *semi-supervised regression* [60], *semi-supervised clustering* such as constrained and seeded k-means clustering [53, 51, 6] and *semi-supervised dimensionality reduction* [7, 62]. In this survey semi-supervised learning refers to semi-supervised classification, where one has additional unlabeled data and the goal is classification. Many semi-supervised classification algorithms have been developed. They can be divided into five categories according to [63]: (1) *Self-Training* [44], (2) semi-supervised learning with generative models [40, 45, 49], (3) S3VMs (Semi-Supervised Support Vector Machines) [31, 18, 27, 35], (4) semi-supervised learning with graphs [8, 56, 64], and (5) semi-supervised learning with committees (semi-supervised by disagreement) [13, 45, 33, 57, 61, 38, 59].

3 Self-Training

Self-Training [44] is an incremental algorithm that initially builds a single classifier using a small amount of labeled data. Then it iteratively predicts the labels of the unlabeled examples, rank the examples by confidence in their prediction and permenantly adds the *most confident examples* into the labeled training set. It retrains the underlying classifier with the augmented training set and the process is repeated for a given number of iterations or until some heuristic convergence criterion is satisfied. The classification accuracy can be improved over iterations only if the initial and subsequent classifiers correctly label most of the unlabeled examples. Unfortunately, adding mislabeling noise is not avoidable. In practical applications, more accurate confidence measures and predefined confidence thresholds are used in order to limit the number of mislabeled examples.

Self-Training is a wrapper algorithm that can be applied to many learning algorithms. It has been appeared in the literature with several names: self-learning [47, 42], self-corrective recognition [42], naive labelling [30], and decision-directed [55]. One drawback when *Self-Training* is applied on linear classifiers such as *support vector machines* is that the most confident examples often lie away from the target decision boundary (non informative examples). Therefore, in many cases this process does not create representative training sets as it selects non informative examples. Note that an example is called informative, if it lies close to the separating hyperplane and therefore it can influence its position. Another drawback is that *Self-Training* is sensitive to outliers.

4 SSL with Generative Models

In generative approaches, it is assumed that both labeled and unlabeled examples come from the same parametric model where the number of components, prior $p(y)$, and conditional $p(x|y)$ are all known and correct. Once the model parameters are learned, unlabeled examples are classified using the mixture components associated to each class. Methods in this category such as in [45, 43] usually treat the class labels of the unlabeled data $\{x_j\}_{j=1}^{u}$ as missing values and employ the *EM* (Expectation-Maximization) algorithm [21] to conduct maximum likelihood estimation (MLE) of the model parameters θ. It begins with an initial model trained on the labeled examples $\{(x_i,y_i)\}_{i=1}^{l}$. It then iteratively uses the current model to temporarily estimate the class probabilities of all the unlabeled examples and then maximizes the likelihood of the parameters (trains a new model) on all labeled examples (the original and the newly labeled) until it converges.

$$\log p(X_l,Y_l,X_u|\theta)=\sum_{i=1}^{l}\log p(y_i|\theta)p(x_i|y_i,\theta)+\lambda\sum_{j=1}^{u}\log(\sum_{y=1}^{2}p(y|\theta)p(x_j|y,\theta)) \quad (1)$$

The methods differ from each other by the generative models used to fit the data, for example, mixture of Gaussian distributions (GMM) is used for image classification [49], mixture of multinomial distributions (Naive Bayes) [45, 43] is used for text categorization and Hidden Markov Models (HMM) [30] is used for speech recognition. Although the generative models are simple and easy to implement and may be more accurate than discriminative models when the amount of labeled examples is very small, the methods in this category suffer from a serious problem. That is, when the model assumption is incorrect, fitting the model using a large amount of unlabeled data will result in performance degradation [19]. Thus, in order to reduce the danger [63], one needs to carefully construct the generative model, for instance to construct more than one Gaussian component per class. Also, one can down weight the unlabeled examples in the maximum likelihood estimation (set $\lambda<1$).

5 Semi-supervised SVMs (*S3VMs*)

Considering that the training set is divided into two disjoint subsets L for labeled data and U for unlabeled data. The aim of *S3VM* learning algorithm is to exploit the abundant unlabeled data $U=\{x_j\}_{j=1}^{u}$ to adjust the decision boundary initially constructed from a small amount of labeled data $L=\{(x_i,y_i)\}_{i=1}^{l}$,$y_i=\pm 1$, such that it goes through the low density regions while keeping the labeled examples correctly classified [31, 18], see Figure 3. The following optimization problem is solved over both the decision boundary parameters (w,b) and a vector of binary labels assigned to unlabeled examples $\hat{y}_U=(\hat{y}_1,\ldots,\hat{y}_u)^T\in\{-1,1\}^u$:

$$\min_{w,b,\hat{y}_U}\Psi(w,b,\hat{y}_U)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{l}V(y_i,f(x_i))+C^*\sum_{j=1}^{u}V(\hat{y}_j,f(x_j)), \quad (2)$$

where $f(x_i) = \langle w, \phi(x_i) \rangle - b$ is the decision function of *SVM*, ϕ is a nonlinear function that maps an input vector x_i into a high-dimensional dot-product feature space where it is possible to construct an optimal separating hyperplane with better generalization ability and V is a margin loss function. The Hinge loss is a popular loss function that is defined as,

$$V(y_i, f(x_i)) = \max(0, 1 - y_i f(x_i))^p \text{ and } V(\hat{y}_j, f(x_j)) = \max(0, 1 - \hat{y}_j f(x_j))^p \quad (3)$$

where p=1 or 2. It is an extension of the standard support vector machines. In the standard *SVM*, only the labeled data is used while in *S3VM* the unlabeled data is also used. The first two terms in the objective function in Eq. (2) define a standard *SVM*. The third term incorporates unlabeled data. The loss over labeled and unlabeled examples is weighted by two parameters, C and C^*, which reflect confidence in class labels and in the cluster assumption respectively. The minimization problem in Eq. (2) is solved under the following class balancing constraint

$$\frac{1}{u} \sum_{j=1}^{u} \max(0, \hat{y}_j) = r \quad (4)$$

This constraint helps to avoid unbalanced solutions by enforcing that a certain user-specified fraction, r, of the unlabeled data should be assigned to the positive class. The minimization techniques of Ψ can be divided into two broad strategies:

1. Combinatorial Optimization: For a given fixed $\hat{y}_U$, find the optimal solution for (w, b) which is the standard SVM training. The goal now is to minimize Γ over a set of binary variables.

$$\Gamma(\hat{y}_U) = \min_{w,b} \Psi(w, b, \hat{y}_U) \quad (5)$$

2. Continuous Optimization: For a given fixed (w, b), find the optimal $\hat{y}_U$. The unknown variables $\hat{y}_U$ are excluded from optimization, which leads to the following continuous objective function over (w, b):

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{l} \max(0, 1 - y_i f(x_i))^2 + C^* \sum_{j=1}^{u} \max(0, 1 - |f(x_j)|)^2 \quad (6)$$

   That can be solved by continuous optimization techniques. The balancing constraint becomes as follows

$$\frac{1}{u} \sum_{j=1}^{u} f(x_j) = r \quad (7)$$

   After optimization, $\hat{y}_U$ is simply found by applying the decision function on the unlabeled example, $\hat{y}_j = arg \min_{y \in \{-1,1\}} V(y, f(x_j)) = sign(f(x_j))$.

Since its first implementation by Joachims [31], the non-convexity of the problem associated with *S3VM* motivates the development of a number of optimization techniques, for instance, local combinatorial search [31], gradient descent [18],

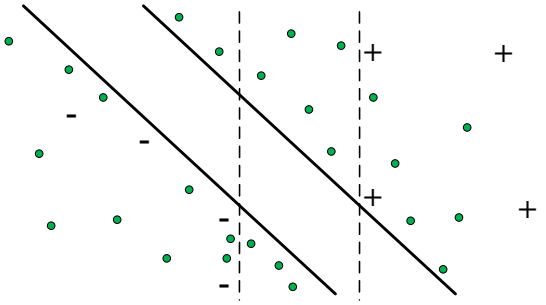


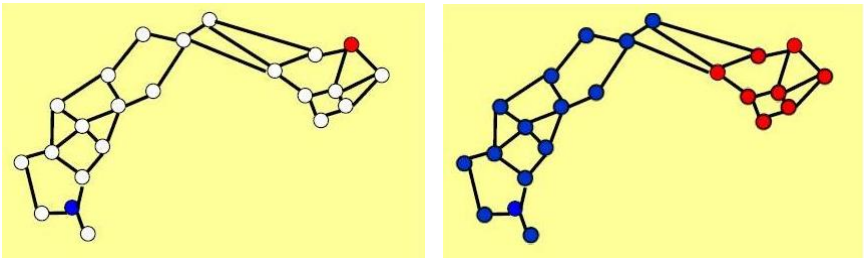
Fig. 3 Graphical illustration of S3VMs: The unlabeled examples help to put the decision boundary in low density regions. Using labeled data only, the maximum margin separating hyperplane is plotted with the versicle dashed lines. Using both labeled and unlabeled data (dots), the maximum margin separating hyperplane is plotted with the oblique solid lines.

continuation techniques [16], convex-concave procedures [25], semi-definite programming [10], deterministic annealing [50], genetic algorithm optimization [4] and branch-and-bound algorithms [17]. In [31], an initial SVM classifier is firstly constructed using the available labeled examples and then the labels of the unlabeled examples y_U^* are iteratively predicted. Then it maximizes the margin over both labeled and the (newly labeled) unlabeled examples $\{(x_j^*, y_j^*)\}_{j=1}^u$. The optimal decision boundary is the one that has the minimum training error on both labeled and unlabeled data. *S3VM*, sometimes called *Transductive SVM*, assumes that unlabeled data from different classes are separated with large margin. In addition, it assumes there is a low density region through which the separating hyperplane passes. Thus, it does not work for domains in which this assumption is not fulfilled.

6 Semi-supervised Learning with Graphs

Blum and Chawla [11] proposed the first graph-based semi-supervised learning method. They constructed a graph whose nodes represent both labeled and unlabeled training examples and the edges between nodes are weighted according to the similarity between the corresponding examples. Based on the graph, the aim is to find the minimum cut of the graph such that nodes in each connected component have the same label. Later, Blum et al. [12] added random noise to the edge weights and the labels of the unlabeled examples are predicted using majority voting. The procedure is similar to bagging and produces a soft minimum cut. Note that in both [11] and [12] a discrete predictive function is used that assigns one of the possible labels to each unlabeled example. Zhu et al. [64] introduced a continuous prediction function. They modeled the distribution of the prediction function over the graph with Gaussian random fields and analytically proved that the prediction function with the lowest energy should have the harmonic property. They designed a label propagation strategy over the graph using such a harmonic property where

the labels propagate from the labeled nodes to the unlabeled ones, see Figure 4. It is worth noting that all graph-based methods assume that examples connected by strong edges tend to have the same class label and vice versa [63]. It is noteworthy that most of the graph-based semi-supervised learning usually focus on how to conduct semi-supervised learning over a given graph. A key that will seriously influence the learning performance is how to construct a graph which reflects the essential similarities among examples.



(a) Before SSL (b) After SSL

Fig. 4 Graphical illustration of label propagation

7 Semi-supervised Learning with Committees (*SSLC*)

The main factor for the success of any committee-based semi-supervised learning, sometimes called semi-supervised learning by disagreement [59], is to construct an ensemble of diverse and accurate classifiers, let them collaborate to exploit unlabeled examples, and maintain a large disagreement (diversity) between these classifiers. In this section, existing committee-based semi-supervised learning techniques are divided into three categories, that is, learning with multiple views and learning with single view multiple classifiers.

7.1 SSLC with Multiple Views

Multi-view learning is based on the assumption that the instance input space $X = X_1 \times X_2$, where $X_1 \subset \mathbb{R}^{D_1}$ and $X_2 \subset \mathbb{R}^{D_2}$ represent two different descriptions of an instance, called views. These views are obtained through different physical sources/sensors or are derived by different feature extraction procedures and are giving different types of discriminating information about the instance. For instance, in visual objective recognition tasks, an image can be described by color, shape or texture. In emotion recognition tasks, an emotion can be recognized from either speech and facial expressions.

Multi-view learning was first introduced for semi-supervised learning by Blum and Mitchell in the context of *Co-Training* [13]. They state two strong requirements for successful *Co-Training*: the two sets of features should be conditionally

independent given the class and either of them sufficient to learn the classification task.. The pseudo-code is shown in Algorithm 1 (see Figure 5). At the initial iteration, two classifiers are trained using a small amount of labeled training data. Then at each further iteration, each classifier predicts the class label of the unlabeled examples, estimates the confidence in its prediction, ranks the examples by confidence, adds the examples about which it is *most confident* into the labeled training set. The aim is that the *most confident* examples with respect to one classifier can be *informative* with respect to the other. An example is informative with respect to a classifier if it carries a new discriminating information. That is, it lies close to the decision boundary and thus adding it to the training set can improve the classification performance of this classifier. Nigam and Ghani [44] showed that *Co-Training* is sensitive to the view independence requirement.

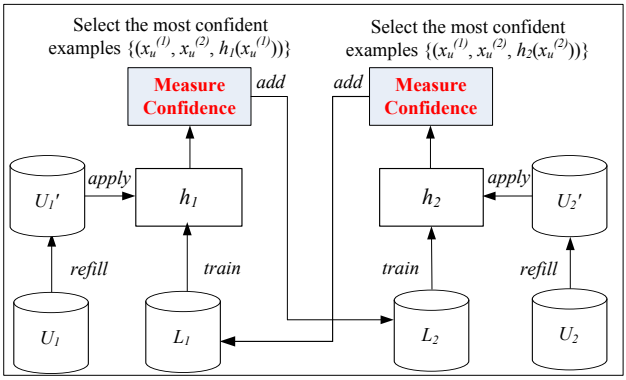


Fig. 5 Graphical illustration of *Co-Training*

Nigam and Ghani [44] proposed another multi-view semi-supervised algorithm, called *Co-EM*. It uses the model learned in one view to probabilistically label the unlabeled examples in the other model. Intuitively, *Co-EM* runs *EM* (Section 4) in each view and before each new *EM* iteration, inter-changes the probabilistic labels predicted in each view. *Co-EM* is considered as a probabilistic variant of *Co-Training*. Both algorithms are based on the same idea: they use the knowledge acquired in one view, in the form of soft class labels for the unlabeled examples, to train the other view. The major difference between the two algorithms is that *Co-EM* does not commit to the labels predicted in the previous iteration because it uses probabilistic labels that may change from one iteration to the other. On the other hand, Co-Training commits to the most confident predictions that are once added into the training set are never revisited. Thus, it may add to the training set a large number of mislabeled examples.

The standard *Co-Training* was applied in domains with truly independent feature splits satisfying its conditions. In [33], Kiritchenko et al. applied *Co-Training* for email classification where the bags of words that represent email messages were split into two sets: the words from headers (V_1) and the words from bodies (V_2).

Algorithm 1. Pseudo code of *Standard Co-Training*

Require: set of labeled training examples (L), set of unlabeled training examples (U), maximum number of iterations (T),base learning algorithm (*BaseLearn*), two feature sets (views) representing an example (V_1,V_2), sample size (n), number of unlabeled examples in the pool (u) and number of classes (C)

Training Phase

1: Get the class prior probabilities, $\{Pr_c\}_{c=1}^{C}$
2: Set the class growth rate, $n_c = n \times Pr_c$ where $c = 1,\ldots,C$
3: Train initial classifiers $h_1^{(0)}$ and $h_2^{(0)}$ on the initial L
$h_1^{(0)} = BaseLearn(V_1(L))$ and $h_2^{(0)} = BaseLearn(V_2(L))$
4: **for** $t \in \{1,\ldots,T\}$ **do**
5: **if** U is empty **then**
6: $T \leftarrow t$-1 and abort loop
7: **end if**
8: **for** $v \in \{1,2\}$ **do**
9: Apply $h_v^{(t-1)}$ on U.
10: Select a subset S_v as follows: for each class ω_c, select the n_c most confident examples assigned to class ω_c
11: Move S_v from U to L
12: **end for**
13: Re-train classifiers $h_1^{(t)}$ and $h_2^{(t)}$ on the new L
$h_1^{(t)} = BaseLearn(V_1(L))$ and $h_2^{(t)} = BaseLearn(V_2(L))$
14: **end for**

Prediction Phase

15: **return** combination of the predictions of $h_1^{(T)}$ and $h_2^{(T)}$

Levin et al. [36] have used *Co-Training* to improve visual detectors for cars in traffic surveillance video where one classifier detects cars in the original gray level images (V_1). The second one uses images where the background has been removed (V_2).

Abdel Hady et al. [3] have combined *Co-Training* with tree-structured approach for multi-class decomposition through two different architectures. In the first architecture, *cotrain-of-trees*, a tree-structured ensemble of binary RBF networks is trained on each given view. Then, using Co-Training the most confident unlabeled examples labeled by each tree ensemble classifier are added to the training set of the other tree classifier. A combination method based on Dempster-Schafer evidence theory provides class probability estimates that were used to measure confidence on prediction. In the second architecture, *tree-of-cotrains*, first the given K-class problem is decomposed into K-1 simpler binary problems using the tree-structured approach. Then using Co-Training a binary RBF network is trained on each given view to solve each binary problem. In order to combine the intermediate results of the internal nodes within each tree, the above mentioned evidence-theoretic combination method is used. Then cotrain-of-trees and tree-of-cotrains were evaluated on three real-world 2D and 3D visual object recognition tasks where one classifier is based on color histograms (V_1) while the second uses orientation histograms (V_2).

Although there are some cases in which there are two or more independent and redundant views, there exist many real-world applications in which multiple views are not available or it is computationally inefficient to extract more than one feature set for each example.

Co-Training was applied in domains without natural feature splits through splitting the available feature set into two views V_1 and V_2. Nigam and Ghani [44] investigated the influence of the views independence. They found that *Co-Training* works better on truly independent views than on random views. Also, *Co-Training* was found to outperform EM when the views are truly independent. It was also shown that if there is sufficient redundancy in data, the performance of *Co-Training* with random splits is comparable to *Co-Training* with a natural split. Of course there is no guarantee that random splitting will produce independent views.

Feger and Koprinska [22] introduced a method, called *maxInd*, for splitting the feature set into two views. The aim is to minimize the dependence between the two feature subsets (*inter-dependence*), measured by conditional mutual information *CondMI*. The result is represented as an undirected graph, with features as nodes and the *CondMI* between each pair of features as weight on the edge between them. In the second step the graph is cut into two disjoint parts of the same size. This split is performed in such a way that minimizes the sum of the cut edges in order to minimize the dependence between the two parts of the graph. They had found that *maxInd* does not outperform the random splits. A possible explanation from their perspective is that *Co-Training* is sensitive to the dependence of the features within each view (*intra-dependence*). The random split leads to *intra-dependence* lower than that of *maxInd* and the truly independent split. Their study states that there is a trade-off between the *intra-dependence* of each view, and the *inter-dependence* between the views. That is minimizing the *inter-dependence* leads to maximizing the *intra-dependence* of each view. In addition, the measurement of *CondMI* is not accurate enough because it is based on only a small number of labeled examples.

Salaheldin and El Gayar [46] introduced three new criteria for splitting features in *Co-Training* and compare them to existing artificial splits and natural split. The first feature split criterion is based on maximizing the confidence of the views. The second criterion maximizes both confidence and independence of the views. The independence of a view is measured by conditional mutual information as in [22]. For each view, a classifier is trained using the labeled data; it is then used to predict the class of the unlabeled data. The entropy of the classifier output for each input example is calculated and the average of entropies indicates the confidence of the view. They showed that splitting the features with a mixed criterion is better than using each criterion alone. Finally, they proposed a third criterion based on maximizing the views diversity. A genetic algorithm is used to optimize the fitness functions based on the three proposed criteria. The experimental results on two data sets show that the proposed splits are promising alternatives to random splitting.

7.2 SSLC with Single View

7.2.1 For Classification

In a number of recent studies [26, 57, 61, 38], the applicability of *Co-Training* using a single view without feature splitting has been investigated. Goldman and Zhou [26] first presented a single-view *SSL* method, called *Statistical Co-learning*. It used two different supervised learning algorithms with the assumption that each of them produce a hypothesis that partition the input space into a set of equivalence classes. For example, a decision tree partitions the input space with one equivalence class per leaf. They used 10-fold cross validation:(1) to select the most confident examples to label at each iteration and (2) to combine the two hypotheses producing the final decision. Its drawbacks are: first the assumptions concering the used algorithms limits its applicability. Second the amount of available labeled data was insufficient for applying cross validation which is time-consuming. Zhou and Goldman [57] then presented another single view method, called *Democratic Co-learning* which is applied to three or more supervised learning algorithms and reduce the need for statistical tests. Therefore, it resolves the drawbacks of *Statistical Co-learning* but it still uses the time-consuming cross-validation technique to measure confidence intervals. These confidence intervals are used to select the most confident unlabeled examples and to combine the hypotheses decisions.

Zhou and Li [61] present a new *Co-Training* style *SSL* method, called *Tri-Training*, where three classifiers are initially trained on bootstrap subsamples generated from the original labeled training set. These classifiers are then refined during the *Tri-Training* process, and the final hypothesis is produced via majority voting. The construction of the initial classifiers looks like training an ensemble from the labeled data with *Bagging* [14]. At each *Tri-Training* iteration, an unlabeled example is added to the training set of a classifier if the other two classifiers agree on their prediction under certain conditions. *Tri-Training* is more applicable than previous *Co-Training*-Style algorithms because it neither requires multiple views as in [13, 44] nor does it depend on different supervised learning algorithms as in [26, 57]. There are two limitations for *Tri-Training*: the ensemble size is limited to three classifiers and it hurts the diversity as *Bagging* is used as ensemble learner. Therefore, the classifiers become identical through the iterations because their training sets become similar. The reason is that the unlabeled examples added to one classifier are not removed from the unlabeled data set therefore the same examples can be selected and added to another classifier at the same iteration or in further iterations.

Li and Zhou [38] proposed an extension to *Tri-Training*, called *Co-Forest*. The aim is to maintain the diversity during the SSL process through using *Random Forest* instead of *Bagging*. That is an initial ensemble of random trees is trained on bootstrap subsamples generated from the given labeled data set L. To select new training examples from a given unlabeled data set U for each ensemble member h_i ($i = 1, \ldots, N$), a new ensemble H_i, called the concomitant ensemble of h_i, is defined that contains all the classifiers except h_i. At each iteration t and for each ensemble member h_i, first the error rate of H_i, $\hat{\varepsilon}_{i,t}$, is estimated. If $\hat{\varepsilon}_{i,t}$ is less than $\hat{\varepsilon}_{i,t-1}$

(1^{th} condition), H_i predicts the class label of the unlabeled examples in $U'_{i,t}$ (random subsample of U of size $\frac{\hat{\varepsilon}_{i,t-1}W_{i,t-1}}{\hat{\varepsilon}_{i,t}}$). A set $L'_{i,t}$ is defined that contains the unlabeled examples in $U'_{i,t}$ where the confidence of H_i about their prediction exceeds a predefined threshold (θ) and $W_{i,t}$ is the sum of the confidences of the examples in $L'_{i,t}$. If $W_{i,t}$ is greater than $W_{i,t-1}$ (2^{nd} condition) and $\hat{\varepsilon}_{i,t}W_{i,t}$ is less than $\hat{\varepsilon}_{i,t-1}W_{i,t-1}$ (3^{rd} condition), the i^{th} random tree will be re-trained using the original labeled data set L and $L'_{i,t}$. Note that the bootstrap sample used to train the i^{th} random tree at iteration 0 is discarded and $L'_{i,t}$ is not added *permenantly* into L. The algorithm will stop if there is no classifier h_i satisfying the three conditions.

d'Alché et al. [20] generalized *MarginBoost* to semi-supervised classification. *MarginBoost* is a variant of AdaBoost [23] based on the minimization of an explicit cost function. Such function is defined for any scalar decreasing function of the margin. As the usual definition of margin cannot be used for unlabeled data, the authors extend the margin notion to unlabeled data. In practice, the margin is estimated using the *MarginBoost* classification output. Then, they reformulate the cost function of *MarginBoost* to include both the labeled and unlabeled data. A generative model is used as a base classifier and the unlabeled data is used by EM algorithms. The results have shown that *SSMBoost* outperforms the classical AdaBoost when a few amount of labeled data is available (only 5% of the training data is labeled).

Bennet et al. [9] proposed another committee-based SSL method, called *ASSEMBLE*, which iteratively constructs ensemble classifiers using both labeled and unlabeled data. The aim of *ASSEMBLE* is to overcome some limitations of *SSMBoost*. For example, while *SSMBoost* requires the base classifier to be a generative mixture model in order to apply EM for semi-supervision, *ASSEMBLE* is more general that can be used with any cost-sensitive base learning algorithm. At each iteration of *ASSEMBLE*, the unlabeled examples are assigning pseudo-classes using the current ensemble before constructing the next base classifier using both the labeled and newly-labeled examples. The experiments show that *ASSEMBLE* works well and it won the NIPS 2001 unlabeled data competition using decision trees as base classifiers.

Abdel Hady and Schwenker [1] introduced a new *committee-based single-view Co-Training* style algorithm, *CoBC*, for application domains in which the available data is not described by multiple redundant and independent views. *CoBC* works as follows: firstly the class prior probabilities are determined then an initial committee of N diverse accurate classifiers $H^{(0)}$ is trained on L using the given ensemble learning algorithm *EnsembleLearn* and base learning algorithm *BaseLearn*. Then the following steps are repeated until the maximum number of iterations T is reached or U becomes empty. For each iteration t and for each classifier i, a set $U'_{i,t}$ of u examples drawn randomly from U without replacement. It is computationally more efficient to use $U'_{i,t}$ instead of using the whole set U. The method *SelectCompetentExamples* (see Algorithm 3) is applied to estimate the competence of each unlabeled example in $U'_{i,t}$ given the companion committee $H_i^{(t-1)}$. Note that $H_i^{(t-1)}$ is the ensemble of all base classifiers trained in the previous iteration except $h_i^{(t-1)}$. A set $\pi_{i,t}$ is created that contains the n_c most competent examples assigned to each class ω_c.

Algorithm 2. Pseudo code of *CoBC* for classification

Require: set of labeled training examples (L), set of unlabeled training examples (U), maximum number of iterations (T), ensemble learning algorithm (*EnsembleLearn*), base learning algorithm (*BaseLearn*), ensemble size (N), number of unlabeled examples in the pool (u), number of nearest neighbors (k), sample size (n), number of classes (C) and an initial committee ($H^{(0)}$)

Training Phase

1: Get the class prior probabilities, $\{Pr_c\}_{c=1}^{C}$
2: Set the class growth rate, $n_c = n \times Pr_c$ where $c = 1,\ldots,C$
3: **if** $H^{(0)}$ is not given **then**
4: Construct an initial committee of N classifiers, $H^{(0)} = EnsembleLearn(L, BaseLearn, N)$
5: **end if**
6: **for** $t \in \{1,\ldots,T\}$ **do**
7: $L'_t \leftarrow \emptyset$
8: **if** U is empty **then** $T = t$-1 and abort loop **end if**
{Get most confident examples ($\pi_{i,t}$) using companion committee $H_i^{(t-1)}$ }
9: **for** $i \in \{1,\ldots,N\}$ **do**
10: $U'_{i,t} \leftarrow RandomSubsample(U, u)$
11: $\pi_{i,t} \leftarrow SelectCompetentExamples(i, U'_{i,t}, H_i^{(t-1)}, k, \{n_c\}_{c=1}^{C}, C)$
12: $L'_t \leftarrow L'_t \cup \pi_{i,t}$, $U'_{i,t} \leftarrow U'_{i,t} \setminus \pi_{i,t}$ and $U \leftarrow U \cup U'_{i,t}$
13: **end for**
14: **if** L'_t is empty **then** $T = t$-1 and abort loop **end if**
{Re-train the N classifiers using their augmented training sets }
15: **for** $i \in \{1,\ldots,N\}$ **do**
16: $L_i = L_i \cup L'_t$
17: $h_i^{(t)} = BaseLearn(L_i)$ (for incremental learning, $h_i^{(t)} = BaseLearn(h_i^{(t-1)}, L'_t)$)
18: **end for**
19: **end for**

Prediction Phase

20: **return** $H^{(T)}(x) = \frac{1}{N}\sum_{i=1}^{N} h_i^{(T)}(x)$ for a given example x

Then $\pi_{i,t}$ is removed from $U'_{i,t}$ and inserted into the set L'_t that contains all the examples labeled at iteration t. The remaining examples in $U'_{i,t}$ are returned to U. There are two options: (1) if the underlying ensemble learner depends on training set perturbation to promote diversity, then insert $\pi_{i,t}$ only into L_i. Otherwise, $h_i^{(t)}$ and $h_j^{(t)}$ $(i \neq j)$ will be identical because they are refined with the same newly labeled examples. This will degrade the ensemble diversity and therefore degrades the relative improvement expected due to exploiting the unlabeled data. One can observe that if the ensemble members are identical, *CoBC* will degenerate to *Self-Training*. (2) If ensemble learner employs another source of diversity, then it is not a problem to insert $\pi_{i,t}$ into the training sets of all classifiers as shown in step 16. Then, *CoBC* does not recall *EnsembleLearn* but only the N committee members are retrained using their updated training sets L_i. It is worth noting that: (1) *CoBC* can improve the recognition rate only if the most confident examples with respect to the

companion committee H_i are informative examples with respect to h_i. (2) Although *CoBC* selects the most confident examples, adding mislabeled examples to the training set (*noise*) is unavoidable but the negative impact of this *noise* could be compensated by augmenting the training set with sufficient amount of newly labeled examples.

A new confidence measure is proposed (Algorithm 3) in order to compensate the inaccurate probability-based ranking provided by traditional decision trees. That is, all unlabeled examples x_u which lie into a particular leaf node (region), will have the same class probability estimates (*CPE*)s because the *CPE* depends on class frequencies and not the distance between x_u and the decision boundaries. The new measure depends on estimating the companion committee accuracy on labeling the neighborhood of an unlabeled example x_u. This local accuracy represents the probability that the companion committee correctly predicts the class label of x_u. The local competence of an unlabeled example x_u given a companion committee $H_i^{(t-1)}$ can be defined as follows:

$$Comp(x_u, H_i^{(t-1)}) = \sum_{\substack{(x_n, y_n) \in N_k(x_u) \\ y_n = \hat{y}_u}} W_n . H_i^{(t-1)}(x_n, \hat{y}_u) \tag{8}$$

where

$$W_n = \frac{1}{||x_n - x_u||_2 + \varepsilon}, \tag{9}$$

$$\hat{y}_u = arg \max_{1 \le c \le C} H_i^{(t-1)}(x_u, \omega_c), \tag{10}$$

$H_i^{(t-1)}(x_n, \hat{y}_u)$ is the probability given by $H_i^{(t-1)}$ that neighbor x_n belongs to the same class assigned to x_u ($\hat{y}_u$), W_n is the reciprocal of the Euclidean distance between x_u and its neighbor x_n and ε is a constant added to avoid zero denominator. The neighborhood could also be determined using a separate validation set (a set of labeled examples that is not used for training the classifiers), but it may be impractical to spend a part from the small-sized labeled data for validation. To avoid the inaccurate estimation of local accuracy that may result due to overfitting, the newly-labeled training examples $\pi_{i,t}$ will not be involved in the estimation. That is, only the initially (manually) labeled training examples are taken into account. Then, the set $N_k(x_u)$ is defined as the set of k nearest labeled examples to x_u.

The local competence assumes that the actual data distribution satisfies the well-known cluster assumption: examples with similar inputs should belong to the same class. Therefore, the local competence of x_u is zero if there is not any neighbor belongs to the predicted class label $\hat{y}_u$ which contradicts the cluster assumption. Therefore, one can observe that $\hat{y}_u$ is an incorrect class label of x_u ($\hat{y}_u \neq y_u$). In addition, the local competence increases as the number of neighbors that belong to $\hat{y}_u$ increases and as the distances between these neighbors and x_u decreases.

Experiments were conducted on ten image recognition tasks in which the random subspace method [28] is used to construct ensembles of diverse 1-nearest neighbor classifiers and C4.5 decision trees. The results verify the effectiveness of *CoBC* to exploit the unlabeled data given a small amount of labeled examples.

Algorithm 3. Pseudo Code of the *SelectCompetentExamples* method

Require: pool of unlabeled examples ($U'_{i,t}$), the companion committee of classifier $h_i^{(t-1)}$ ($H_i^{(t-1)}$), number of nearest neighbors k, growth rate ($\{n_c\}_{c=1}^{C}$) and number of classes (C)

$\pi_{i,t} \leftarrow \emptyset$
for each class $\omega_c \in \{\omega_1, \ldots, \omega_C\}$ **do**
  $count_c \leftarrow 0$
end for
for each $x_u \in U'_{i,t}$ **do**
  $H_i^{(t-1)}(x_u) = \frac{1}{N-1} \sum_{j=1,\ldots,N, j \neq i} h_j^{(t-1)}(x_u)$
  Apply the companion committee $H_i^{(t-1)}$ to x_u,
  $\hat{y}_u \leftarrow arg\max_{1 \leq c \leq C} H_i^{(t-1)}(x_u, \omega_c)$
  Find the k nearest neighbors of x_u,
  $N_k(x_u) = \{(x_n, y_n) | (x_n, y_n) \in Neighbors(x_u, k, L)\}$
  Calculate $Comp(x_u, H_i^{(t-1)})$ as defined in Eq. (8) and Eq. (9)
end for
Rank the examples in $U'_{i,t}$ based on competence (in descending order)
{Select the n_c examples with the maximum competence for class ω_c}
for each $x_u \in U'_{i,t}$ **do**
  if $Comp(x_u, H_i^{(t-1)}) > 0$ and $count_{\hat{y}_u} < n_{\hat{y}_u}$ **then**
    $\pi_{i,t} = \pi_{i,t} \cup \{(x_u, \hat{y}_u)\}$ and $count_{\hat{y}_u} = count_{\hat{y}_u} + 1$
  end if
end for
return $\pi_{i,t}$

7.2.2 For Regression

Previous studies on semi-supervised learning mainly focus on classification tasks. Although regression is almost as important as classification, semi-supervised regression has rarely been studied. One reason is that for real-valued outputs the cluster assumption is not applicable. Although methods based on manifold assumption can be extended to regression, as pointed out by [63], these methods are essentially transductive instead of really semi-supervised since they assume that the unlabeled examples are exactly test examples.

Abdel Hady et al. [2] introduced an extension of *CoBC* for regression, *CoBCReg*. There are two potential problems that can prevent any *Co-Training* style algorithm from exploiting the unlabeled data to improve the performance and these problems are the motivation for *CoBCReg*. Firstly the outputs of unlabeled examples may be incorrectly estimated by a regressor. This leads to adding noisy examples to the training set of the other regressor and therefore *SSL* will degrade the performance. Secondly there is no guarantee that the newly-predicted examples selected by a regressor as *most confident examples* will be *informative examples* for the other regressor. In order to mitigate the former problem, a committee of predictors is used

in *CoBCReg* to predict the unlabeled examples instead of a single predictor. For the latter problem, each regressor selects the most informative examples for itself.

Let L and U represent the labeled and unlabeled training set respectively, which are drawn randomly from the same distribution where for each instance x_μ in L is associated with the target real-valued output while the real-valued outputs of examples in U are unknown. The pseudo-code of *CoBCReg* is shown in Algorithm 4. *CoBCReg* works as follow: initially an ensemble consists of N regressors, which is denoted by H, is constructed from L using *Bagging*. Then the following steps will be repeated until the maximum number of iterations T is reached or U becomes empty. For each iteration t and for each ensemble member h_i, a set U' of u examples is drawn randomly from U without replacement. It is computationally more efficient to use a pool U' instead of using the whole set U. The *SelectRelevantExamples* method (Algorithm 5) is applied to estimate the relevance of each unlabeled example in U' given the *companion committee* H_i. H_i is the ensemble consisting of all member regressors except h_i. A set π_j is created that contains the gr most relevant examples. Then π_j is removed from U' and inserted into the training set of h_i (L_i) such that h_i is refined using the augmented training set L_i. In the prediction phase, the regression estimate for a given example is the weighted average of the outputs of the N regressors created at the final *CoBCReg* iteration. The combination of an ensemble of regressors is only effective if they are diverse. Clearly, if they are identical, then for each regressor, the outputs estimated by the other regressors will be the same as these estimated by the regressor for itself. That is, there is no more knowledge to be transfered among regressors. In *CoBCReg*, there are three sources for diversity creation, the *RBF* network regressors are trained using: (1) different bootstrap samples, (2) different random initialization of *RBF* centers and (3) different distance measures. The Minkowski distance between two D-dimensional feature vectors x_1 and x_2, as defined in Eq. (11), is used with different distance order p to train different *RBF* network regressors. In general, the smaller the order, the more robust the resulting distance metric to data variations. Another benefit of this setting, is that, since it is difficult to find in advance the best p value for a given task, then regressors based on different p values might show complementary behavior.

$$\|x_1 - x_2\|_p = \left(\sum_{i=1}^{D} |x_{1i} - x_{2i}|^p \right)^{1/p} \tag{11}$$

Unlike *Co-Forest* [38], *CoBCReg* does not hurt the diversity among regressors because the examples selected by a regressor are removed from U. Thus, they can not be selected further by other regressors which keeps the training sets of regressors not similar. Even if the training sets become similar, the regressors could still be diverse because they are instantiated with different distance measures, for some data sets this acts like using different feature spaces.

The main challenge for *CoBCReg* is the mechanism for estimating the confidence because the number of possible predictions in regression is unknown. For regression, in [34], variance is used as an effective selection criterion for active learning because a high variance between the estimates of the ensemble members leads to a

Algorithm 4. Pseudo Code of CoBC for Regression

Require: set of l labeled training examples (L), set of u unlabeled examples (U), maximum number of Co-Training iterations (T), *ensemble size* (N), pool size (u), growth rate (gr), number of RBF hidden nodes (k), RBF width parameter (α), distance order of the i^{th} regressor (p_i)

Training Phase

for $i = 1$ to N **do**
  $\{L_i, V_i\} \leftarrow BootstrapSample(L)$ $\{L_i$ is bag and V_i is out-of-bag$\}$
  $h_i = RBFNN(L_i, k, \alpha, p_i)$
end for
for $t \in \{1 \ldots T\}$ **do**
  if U is empty **then** $T = t$-1 and abort loop **end if**
  for $i \in \{1 \ldots N\}$ **do**
    Create a pool U' of u examples by random sampling from U
    $\pi_i = SelectRelevantExamples(i, U', V_i, gr)$
    $U' = U' \setminus \pi_i$ and $U = U \cup U'$
  end for
  for $i \in \{1 \ldots N\}$ **do**
    if π_i is not empty **then**
      $L_i = L_i \cup \pi_i$
      $h_i = RBFNN(L_i, k, \alpha, p_i)$
    end if
  end for
end for

Prediction Phase

return $H(x) = \sum_{i=1}^{N} w_i h_i(x)$ for a given sample x

high average error. Unfortunately, a low variance does not necessarily imply a low average error. That is, it can not be used as a selection criterion for *SSL* because agreement of committee members does not imply that the estimated output is close to the target output. In fact, we will not measure the *labeling confidence* but we will provide another confidence measure called *selection confidence* (See Algorithm 5).

The most relevantly selected example should be the one which minimizes the regressor error on the validation set. Thus, for each regressor h_j, create a pool U' of u unlabeled examples. Then, the root mean squared error (*RMSE*) of h_j is evaluated first (ε_j). Then for each example x_u in U', h_j is refined with $(x_u, H_j(x_u))$ creating new regressor h'_j. So the *RMSE* of h'_j can be evaluated (ε'_j), where $H_j(x_u)$ is the real-valued output estimated by the *companion committee* of h_j (H_j denotes all other ensemble members in H except h_j). Finally, the unlabeled example $\tilde{x}_j$ which maximizes the relative improvement of the *RMSE* (Δ_{x_u}) is selected as the most relevant example labeled by *companion committee* H_j.

It is worth mentioning that the *RMSE*s ε_j and ε'_j should be estimated accurately. If the training data of h_j is used, this will under-estimate the *RMSE*. Fortunately, since the bootstrap sampling [14] is used to construct the committee, the *out-of-bootstrap* examples are considered for a more accurate estimate of ε'_j.

Algorithm 5. Pseudo Code of of the *SelectRelevantExamples* method

Require: the index of the regressor excluded from the committee (j), pool of u unlabeled examples (U'), validation set (V_j), growth rate (gr)

Calculate validation error of h_j using V_j, ε_j
for each $x_u \in U'$ **do**
  $H_j(x_u) = \frac{1}{N-1} \sum_{i=1, i \neq j}^{N} h_i(x_u)$
  $h'_j = RBFNN(L_j \cup \{(x_u, H_j(x_u))\}, k, \alpha, p_j)$
  Calculate validation error ε'_j of h'_j using V_j, then $\Delta_{x_u} = (\varepsilon_j - \varepsilon'_j)/\varepsilon_j$
end for
$\pi_j \leftarrow \emptyset$
for gr times **do**
  if there exists $x_u \in U' \setminus \pi_j$ with $\Delta_{x_u} > 0$ **then**
    $\tilde{x}_j = arg\max_{x_u \in U' \setminus \pi_j} \Delta_{x_u}$
    $\pi_j = \pi_j \cup \{(\tilde{x}_j, H_j(\tilde{x}_j))\}$
  end if
end for
return π_j

8 Combination with Active Learning

Both semi-supervised learning and active learning tackle the same problem but from different directions. That is, they aim to improve the generalization error and at the same time minimize the cost of data annotation through exploiting the abundant unlabeled data.

8.1 SSL with Graphs

Zhu et al. [65] combine *semi-supervised learning* and *active learning* under a Gaussian random field model. Labeled and unlabeled data are represented as nodes in a weighted graph, with edge weights encoding the similarity between examples. Then the semi-supervised learning problem is formulated, in another work by the same authors [64], in terms of a Gaussian random field on this graph, the mean of which is characterized in terms of harmonic functions. *Active learning* was performed on top of the *semi-supervised learning* scheme by greedily selecting queries from the unlabeled data to minimize the estimated expected classification error (risk); in the case of Gaussian fields the risk is efficiently computed using matrix methods. They present experimental results on synthetic data, handwritten digit recognition, and text classification tasks. The active learning scheme requires a much smaller number of queries to achieve high accuracy compared with random query selection. Hoi et al. [29] proposed a novel framework that combine support vector machines and semi-supervised active learning for image retrieval. It is based on the Gaussian fields and harmonic functions semi-supervised approach proposed by Zhu et al. [64].

8.2 SSL with Generative Models

McCallum and Nigam [39] present a Bayesian probabilistic framework for text classification that reduces the need for labeled training documents by taking advantage of a large pool of unlabeled documents. First they modified the *Query-by-Committee* method of active learning (*QBC*) to use the unlabeled pool for explicitly estimating document density when selecting examples for labeling. Then the modified *QBC* is combined with *Expectation-Maximization* (*EM*) in order to predict the class labels of those documents that remain unlabeled. They proposed two approaches to combine *QBC* and *EM*, called *QBC-then-EM* and *QBC-with-EM*. *QBC-then-EM* runs *EM* to convergence after actively selecting all the training examples that will be labeled. This means to use *QBC* to select a better starting point for *EM* hill climbing, instead of randomly selecting documents to label for the starting point. *QBC-with-EM* is a more interesting approach to interleave *EM* with *QBC* so that *EM* not only builds on the results of *QBC*, but *EM* also informs *QBC*. To do this, *EM* runs to convergence on each committee member before performing the disagreement calculations. The aim is (1) to avoid requesting labels for examples whose label can be reliably predicted by EM, and (2) to encourage the selection of examples that will help *EM* find a local maximum likelihood with higher classification accuracy. This directs *QBC* to pick more informative documents to label because it has more accurate committee members. Experimental results show that using the combination of *QBC* and *EM* performs better than using either individually and requires only slightly half the number of labeled training examples required by either *QBC* or EM alone to achieve the same accuracy.

8.3 SSL with Committees

Muslea et al. [41] combined *Co-Testing* and *Co-EM* in order to produce an active multi-view semi-supervised algorithm, called *Co-EMT*. The experimental results on web page classification show that *Co-EMT* outperforms other non-active multi-view algorithms (*Co-Training* and *Co-EM*) without using more labeled data and it is more robust to the violation of the requirements of independent and redundant views.

Zhou et al. [58] proposed an approach, called *SSAIR* (Semi-Supervised Active Image Retrieval), that attempts to exploit unlabeled data to improve the performance of content-based image retrieval (*CBIR*). In detail, in each iteration of relevance feedback, two simple classifiers are trained from the labeled data, i.e. images result from user query and user feedback. Each classifier then predicts the class labels of the unlabeled images in the database and passes the most relevant/irrelevant images to the other classifier. After re-training with the additional labeled data, the classifiers classify the images in the database again and then their classifications are combined. Images judged to be relevant with high confidence are returned as the retrieval result, while these judged with low confidence are put into the pool which is used in the next iteration of relevance feedback. Experiments show that semi-supervised learning and active learning mechanisms are both beneficial to *CBIR*. It is worth mentioning that *SSAIR* depends on single-view versions of *Co-Testing* and

Co-Training that require neither two independent and redundant views nor two different supervised learning algorithm. In order to create the diversity, the two classifiers used for *Co-Testing* and *Co-Training* are trained using the Minkowsky distance metric with different distance order.

Abdel Hady and Schwenker [1] introduced two new approaches, *QBC-then-CoBC* and *QBC-with-CoBC*, that combine the merits of *committee-based active learning* and *committee-based semi-supervised learning*. The first approach is the most straightforward way of combining *CoBC* and active learning where *CoBC* is run after active learning completes (denoted by *QBC-then-CoBC*). The objective is that active learning can help *CoBC* through providing it with a better starting point instead of randomly selecting examples to label for the starting point. A more interesting approach, denoted *QBC-with-CoBC*, is to interleave *CoBC* with *QBC*, so that *CoBC* not only runs on the results of active learning, but *CoBC* also helps *QBC* in the sample selection process as it augments the labeled training set with the most competent examples selected by *CoBC*. Thus, mutual benefit can be achieved. Experiments were conducted on the ten image recognition tasks. The results have shown that both *QBC-then-CoBC* and *QBC-with-CoBC* can enhance the performance of *standalone QBC* and *standalone CoBC*. Also they outperform other non committee-based combinations of semi-supervised and active learning algorithms such that *US-then-ST*, *US-then-CoBC* and *QBC-then-ST*.

9 Conclusion

During the past decade, many semi-supervised learning approaches have been introduced, many theoretical supports have been discovered, and many successful real-world applications have been reported. The work in [13, 5] has theoretically studied *Co-Training* with two views, but could not explain why the single-view variants can work. Wang and Zhou [54] provided a theoretical analysis that emphasizes that the important factor for the success of single-view committee-based *Co-Training style* algorithms is the creation of a large diversity (disagreement) among the co-trained classifiers, regardless of the method used to create diversity, for instance through: sufficiently redundant and independent views as in standard *Co-Training* [13, 44], artificial feature splits in [22, 46], different supervised learning algorithms as in [26, 57], training set manipulation as in [9, 61], different parameters of the same supervised learning algorithms [60] or feature set manipulation as in [38, 1].

Brown et al. presented in [15] an extensive survey of the various techniques used for creating diverse ensembles, and categorized them, forming a preliminary taxonomy of diversity creation methods. One can see that multi-view *Co-Training* is a special case of semi-supervised learning with committees. Therefore, the data mining community is interested in a more general *Co-Training style* framework that can exploit the diversity among the members of an ensemble for correctly predicting the unlabeled data in order to boost the generalization ability of the ensemble.

There is no *SSL* algorithm that is the best for all real-world data sets. Each *SSL* algorithm has its strong assumptions because labeled data is scarce and there is no

guarantee that unlabeled data will always help. One should use the method whose assumptions match the given problem. Inspired by [63], we have the following checklist: If the classes produce well clustered data, then EM with generative mixture models may be a good choice; If the features are naturally divided into two or more redundant and independent sets of features, then standard *Co-Training* may be appropriate; If *SVM* is already used, then *Transductive SVM* is a natural extension; In all cases, *Self-Training* and *CoBC* are practical wrapper methods.

References

1. Abdel Hady, M.F., Schwenker, F.: Combining committee-based semi-supervised learning and active learning. Journal of Computer Science and Technology (JCST): Special Issue on Advances in Machine Learning and Applications 25(4), 681–698 (2010)
2. Abdel Hady, M.F., Schwenker, F., Palm, G.: Semi-supervised Learning for Regression with Co-training by Committee. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009, Part I. LNCS, vol. 5768, pp. 121–130. Springer, Heidelberg (2009)
3. Abdel Hady, M.F., Schwenker, F., Palm, G.: Semi-supervised learning for tree-structured ensembles of RBF networks with co-training. Neural Networks 23(4), 497–509 (2010)
4. Adankon, M., Cheriet, M.: Genetic algorithm–based training for semi-supervised svm. Neural Computing and Applications 19, 1197–1206 (2010)
5. Balcan, M.-F., Blum, A., Yang, K.: Co-Training and expansion: Towards bridging theory and practice. In: Advances in Neural Information Processing Systems 17, pp. 89–96 (2005)
6. Basu, S., Banerjee, A., Mooney, R.: Semi-supervised clustering by seeding. In: Proc. of the 19th International Conference on Machine Learning (ICML 2002), pp. 19–26 (2002)
7. Basu, S., Bilenko, M., Mooney, R.: A probabilistic framework for semi-supervised clustering. In: Proc. of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2004), pp. 59–68 (2004)
8. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research 7, 2399–2434 (2006)
9. Bennet, K., Demiriz, A., Maclin, R.: Exploiting unlabeled data in ensemble methods. In: Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 289–296 (2002)
10. De Bie, T., Cristianini, N.: Semi-supervised learning using semi-definite programming. In: Semi-supervised Learning. MIT Press (2006)
11. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts. In: Proc. of the 18th International Conference on Machine Learning (ICML 2001), pp. 19–26 (2001)
12. Blum, A., Lafferty, J., Rwebangira, M., Reddy, R.: Semi-supervised learning using randomized mincuts. In: Proc. of the 21st International Conference on Machine Learning (ICML 2004), pp. 13–20 (2004)
13. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proc. of the 11th Annual Conference on Computational Learning Theory (COLT 1998), pp. 92–100. Morgan Kaufmann (1998)
14. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)

15. Brown, G., Wyatt, J., Harris, R., Yao, X.: Diversity creation methods: a survey and categorisation. Information Fusion 6(1), 5–20 (2005)
16. Chapelle, O., Chi, M., Zien, A.: A continuation method for semi-supervised svms. In: International Conference on Machine Learning (2006)
17. Chapelle, O., Sindhwani, V., Keerthi, S.: Branch and bound for semi-supervised support vector machines. In: Advances in Neural Information Processing Systems (2006)
18. Chapelle, O., Zien, A.: Semi-supervised learning by low density separation. In: Proc. of the 10th International Workshop on Artificial Intelligence and Statistics, pp. 57–64 (2005)
19. Cozman, F.G., Cohen, I.: Unlabeled data can degrade classification performance of generative classifiers. In: Proc. of the 15th International Conference of the Florida Artificial Intelligence Research Society (FLAIRS), pp. 327–331 (2002)
20. d'Alché-Buc, F., Grandvalet, Y., Ambroise, C.: Semi-supervised MarginBoost. In: Neural Information Processing Systems Foundation, NIPS 2002 (2002)
21. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
22. Feger, F., Koprinska, I.: Co-training using RBF nets and different feature splits. In: Proc. of the International Joint Conference on Neural Networks (IJCNN 2006), pp. 1878–1885 (2006)
23. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
24. Freund, Y., Seung, H.S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. Machine Learning 28, 133–168 (1997)
25. Fung, G., Mangasarian, O.: Semi-supervised support vector machines for unlabeled data classification. Optimization Methods and Software 15, 29–44 (2001)
26. Goldman, S., Zhou, Y.: Enhancing supervised learning with unlabeled data. In: Proc. of the 17th International Conference on Machine Learning (ICML 2000), pp. 327–334 (2000)
27. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. Advances in Neural Information Processing Systems 17, 529–536 (2005)
28. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
29. Hoi, S.C.H., Lyu, M.R.: A semi-supervised active learning framework for image retrieval. In: Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 302–309 (2005)
30. Inoue, M., Ueda, N.: Exploitation of unlabeled sequences in hidden markov models. IEEE Transactions On Pattern Analysis and Machine Intelligence 25(12), 1570–1581 (2003)
31. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proc. of the 16th International Conference on Machine Learning, pp. 200–209 (1999)
32. Kemp, T., Waibel, A.: Unsupervised training of a speech recognizer: Recent experiments. In: Proc. EUROSPEECH, pp. 2725–2728 (1999)
33. Kiritchenko, S., Matwin, S.: Email classification with co-training. In: Proc. of the 2001 Conference of the Centre for Advanced Studies on Collaborative research (CASCON 2001), pp. 8–19. IBM Press (2001)
34. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. Advances in Neural Information Processing Systems 7, 231–238 (1995)

35. Lawrence, N.D., Jordan, M.I.: Semi-supervised learning via gaussian processes. Advances in Neural Information Processing Systems 17, 753–760 (2005)
36. Levin, A., Viola, P., Freund, Y.: Unsupervised improvement of visual detectors using co-training. In: Proc. of the International Conference on Computer Vision, pp. 626–633 (2003)
37. Lewis, D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In: Proc. of the 11th International Conference on Machine Learning (ICML 1994), pp. 148–156 (1994)
38. Li, M., Zhou, Z.-H.: Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. IEEE Transactions on Systems, Man and Cybernetics- Part A: Systems and Humans 37(6), 1088–1098 (2007)
39. McCallum, A.K., Nigam, K.: Employing EM and pool-based active learning for text classification. In: Proc. of the 15th International Conference on Machine Learning (ICML 1998), pp. 350–358. Morgan Kaufmann (1998)
40. Miller, D.J., Uyar, H.S.: A mixture of experts classifier with learning based on both labelled and unlabelled data. Advances in Neural Information Processing Systems 9, 571–577 (1997)
41. Muslea, I., Minton, S., Knoblock, C.A.: Active + semi-supervised learning = robust multi-view learning. In: Proc. of the 19th International Conference on Machine Learning (ICML 2002), pp. 435–442 (2002)
42. Nagy, G., Shelton, G.L.: Self-corrective character recognition systems. IEEE Transactions on Information Theory, 215–222 (1966)
43. Nigam, K.: Using Unlabeled Data to Improve Text Classification. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA (2001)
44. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: Proc. of the 9th International Conference on Information and Knowledge Management, New York, NY, USA, pp. 86–93 (2000)
45. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Machine Learning 39(2-3), 103–134 (2000)
46. Salaheldin, A., El Gayar, N.: New Feature Splitting Criteria for Co-training Using Genetic Algorithm Optimization. In: El Gayar, N., Kittler, J., Roli, F. (eds.) MCS 2010. LNCS, vol. 5997, pp. 22–32. Springer, Heidelberg (2010)
47. Seeger, M.: Learning with labeled and unlabeled data. Technical report, University of Edinburgh, Institute for Adaptive and Neural Computation (2002)
48. Settles, B.: Active learning literature survey. Technical report, Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI (2009)
49. Shahshahani, B., Landgrebe, D.: The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon. IEEE Transactions on Geoscience and Remote Sensing 32(5), 1087–1095 (1994)
50. Sindhwani, V., Keerthi, S., Chapelle, O.: Deterministic annealing for semi-supervised kernel machines. In: International Conference on Machine Learning (2006)
51. Tang, W., Zhong, S.: Pairwise constraints-guided dimensinality reduction. In: Proc. of the SDM 2006 Workshop on Feature Selection for Data Mining (2006)
52. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (1995)
53. Wagstaff, K., Cardie, C., Schroedl, S.: Constrained k-means clustering with background knowledge. In: Proc. of the 18th International Conference on Machine Learning (ICML 2001), pp. 577–584 (2001)
54. Wang, W., Zhou, Z.-H.: Analyzing Co-training Style Algorithms. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 454–465. Springer, Heidelberg (2007)

55. Young, T.Y., Farjo, A.: On decision directed estimation and stochastic approximation. IEEE Transactions on Information Theory, 671–673 (1972)
56. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. Advances in Neural Information Processing Systems 16, 753–760 (2004)
57. Zhou, Y., Goldman, S.: Democratic co-learning. In: Proc. of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004), pp. 202–594. IEEE Computer Society, Washington, DC (2004)
58. Zhou, Z.-H., Chen, K.-J., Jiang, Y.: Exploiting Unlabeled Data in Content-Based Image Retrieval. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 525–536. Springer, Heidelberg (2004)
59. Zhou, Z.-H., Li, M.: Semi-supervised learning by disagreement. Knowledge and Information Systems (in press)
60. Zhou, Z.-H., Li, M.: Semi-supervised regression with co-training. In: Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), pp. 908–913 (2005)
61. Zhou, Z.-H., Li, M.: Tri-training: Exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering 17(11), 1529–1541 (2005)
62. Zhou, Z.-H., Zhang, D., Chen, S.: Semi-supervised dimensionality reduction. In: Proc. of the 7th SIAM International Conference on Data Mining (SDM 2007), pp. 629–634 (2007)
63. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530 (2008)
64. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proc. of the 20th International Conference on Machine Learning (ICML 2003), pp. 912–919 (2003)
65. Zhu, X., Lafferty, J., Ghahramani, Z.: Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In: Proc. of the ICML 2003 Workshop on The Continuum from Labeled to Unlabeled Data (2003)