Tri-Training: Exploiting Unlabeled Data Using Three Classifiers

Abstract – In many practical data mining applications such as web page classification, unlabeled training examples are readily available but labeled ones are fairly expensive to obtain. Therefore, semi-supervised learning algorithms such as co-training have attracted much attention. In this paper, a new co-training style semi-supervised learning algorithm named tri-training is proposed. This algorithm generates three classifier from the original labeled example set. These classifiers are then refined using unlabeled examples in the tri-training process. In detail, in each round of tri-training, an unlabeled example is labeled for a classifier if the other two classifiers agree on the labeling, under certain conditions. Since tri-training neither requires the instance space be described with sufficient and redundant views nor does it put any constraints on the supervised learning algorithm, its applicability is broader than that of previous co-training style algorithms. Experiments on UCI data sets and application to the web page classification task indicate that tri-training performance.

Index Terms – Data Mining, Machine Learning, Learning from Unlabeled Data, Semi-supervised Learning, Co-training, Tri-training, Web Page Classification

I. INTRODUCTION

IN many practical data mining applications such as web page classification, unlabeled training examples are readily available but labeled ones are fairly expensive to obtain because they require human effort. Therefore, semi-supervised learning that exploits unlabeled examples in addition to labeled ones has become a hot topic.

Many current semi-supervised learning algorithms use a generative model for the classifier and employ Expectation Maximization (EM) to model the label estimation or parameter estimation process. For example, mixture of Gaussians, mixture of experts, and naive Bayes have been respectively used as the generative model, while EM is used to combine labeled and unlabeled data for classification. There are also many other algorithms such as using transductive inference for support vector machines to optimize performance on a specific test set, constructing a graph on the examples such that minimum cut on the graph yields an optimal labeling of the unlabeled examples according to certain optimization functions, etc.

A prominent achievement in this area is the co-training paradigm proposed by Blum and Mitchell, which trains two classifiers separately on two different views, i.e. two independent sets of attributes, 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 performed word sense disambiguation by constructing a sense classifier using the local context of the word and a classifier based on the sensed of other occurrences of that word in the same document; Riloff and Jones 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 performed named entity classification using both the spelling of the entity itself and the context in which the entity occurs. It is noteworthy that the co-training paradigm has already been used in many domains such as statistical parsing and noun phrase identification.

The standard co-training algorithm requires two sufficient and redundant views, that is, the attributes be naturally partitioned into two sets, each of which is sufficient for learning and conditionally independent to the other given the class label. Dasgupta et al. have shown that when the requirement is met, the co-trained classifiers could make fewer generalization errors by maximizing their agreement over the unlabeled data. Unfortunately, such a requirement can hardly be met in most scenarios. Goldman and Zhou proposed an algorithm which does not exploit attribute partition. However, it requires using two different supervised learning algorithms that partition the instance space into a set of equivalence classes, and employing time-consuming cross validation technique to determine how to label the unlabeled examples and how to product the final hypothesis.

In this paper, a new co-training style algorithm named tri-training is proposed. Tri-training does not require sufficient and redundant views, nor does it require the use of different supervised learning algorithms whose hypothesis partitions the instance space into a set of equivalence classes. Therefore it can be easily applied to common data mining scenarios. In contrast to previous algorithms that utilize two classifiers, tri-training uses three classifiers. This setting tackles the problem of determining how to label the unlabeled examples and how to produce the final hypothesis, which contributes

much to the efficiency of the algorithm. Moreover, better generalization ability can be achieved through combining these three classifiers. Experiments on UCI data sets and application to the web page classification task show that tri-training can effectively exploit unlabeled data, and the generalization ability of its final hypothesis is quite good, sometimes even outperforms that of the ensemble of three classifiers being provided with labels of all the unlabeled examples.

II. TRI-TRAINING

Let $|U'|$ denote the labeled example set with size and denote the unlabeled example set with size . In previous co-training style algorithms, two classifiers are initially trained from , each of which is then re-trained with the help of unlabeled examples that are labeled by the latest version of the other classifier. In order to determine which example in should be labeled and which classifier should be biased in prediction, the confidence of the labeling of each classifier must be explicitly measured. Sometimes such a measuring process is quite time-consuming.

Assume that besides these two classifiers, i.e. and , a classifier h_1 is initially trained from . Then, for any classifier, an unlabeled example can be labeled for it as long as the other two classifiers agree on the labeling of this example, while the confidence of the labeling of the classifiers are not needed to be explicitly measured. For instance, if and agree on the labeling of an example in , then can be labeled for . It is obvious that in such a schema if the prediction of and on is correct; otherwise will get an example with noisy label. However, even in the worse case, the increase in the classification noise rate can be compensated if the amount of newly labeled example is sufficient, under certain conditions, as shown below.

Inspired by Goldman and Zhou, the finding of Angluin and Laird is used in the following analysis. That is, if a sequence of samples is drawn, where the sample size m satisfies Eq. 1:

$$m \geq \frac{2}{\epsilon^2(1-2\eta)^2} ln(\frac{2N}{\delta}) \tag{1}$$

where is the hypothesis worst-case classification error rate, is an upper bound on the classification noise rate, is the number of hypothesis, and is the confidence, then a

$H_i < 0.5)$
hypothesis that minimizes disagreement with will have the PAC property:

$$Pr[d(H_i, H^*) \geq \epsilon] \leq \delta \tag{1}$$

where is sum over the probability of elements from the symmetric difference between the two hypothesis sets and (the ground-truth). Let where makes Eq. 1 hold equality, then Eq. 1 becomes Eq. 3:

$$\mu = 2\mu ln(\frac{2N}{\delta})$$
$$m = \frac{c}{\epsilon^2(1-2\mu)^2} \tag{1}$$

To simplify the computation, it is helpful to compute the quotient of the constant divided by the square of the error:

$$u = \frac{c}{\epsilon^2} = m(1-2\mu)^2 \tag{1}$$

In each round of tri-training, the classifiers h_2 and h_3 choose some examples in U to label for h_1. Since the classifiers are refined in the tri-training process, the amount as well as the concrete unlabeled examples chosen to label may be different in different rounds. Let L^t and L^{t-1} denote the set of examples that are labeled for h_1 in the t-th round and the $(t-1)$-th round, respectively. Then the training set for h_1 in the t-th round and the $(t-1)$-th round are respectively $L \cup L^t$ and $L \cup L^{t-1}$, whose sample size m^t and m^{t-1} are $|L \cup L^t|$ and $|L \cup L^{t-1}|$, respectively. Note that the unlabeled examples labeled in the $(t-1)$-th round, i.e. L^{t-1}, won't be put into the original labeled example set, i.e. L. Instead, in the t-th round all the examples in L^{t-1} will be regarded as unlabeled and put into U again.

Let η_L denote the classification noise rate of L, that is, the number of examples in L that are mislabeled is $\eta_L|L|$. Let $\check{e}_1^t$ denote the upper bound of the classification error rate of $h_2 \& h_3$ in the t-th round, i.e. the error rate of the hypothesis derived from the combination of h_2 and h_3. Assuming there are z number of examples on which the classification made by h_2 agrees with that made by h_3, and among these examples both h_2 and h_3 make correct classification on z' examples, then $\check{e}_1^t$ can be estimated as $\frac{(z-z')}{z}$. Thus, the number of examples in L^t that are mislabeled is $\check{e}_1^t|L^t|$. Therefore the classification noise rate in the t-th round is:

$$\eta^t = \frac{\eta_L|L| + \check{e}_1^t|L^t|}{|L \cup L^t|} \tag{1}$$

Then, according to Eq, u^t can be computed as:

$$u^t = m^t(1-2\eta^t)^2 = |L \cup L^t|\big(1 - 2\frac{\eta_L|L| + \check{e}_1^t|L^t|}{|L \cup L^t|}\big)^2 \quad (2)$$

The pseudo-code of tri-training is presented in Table I. The function $MeasureError(h_j h_k)$ attempts to estimate the classification error rate of the hypothesis derived from the combination of h_j and h_k. Since it is difficult to estimate the classification error on the unlabeled examples, here only the original labeled examples are used, heuristically based on the assumption that the unlabeled examples hold the same distribution as that held by the labeled ones. In detail, the classification error of the hypothesis is approximated through dividing the number of labeled examples on which both h_j and h_k make incorrect classification by the number of labeled examples on which the classification made by h_j is the same as that made by h_k. The function $Subsample(L^t, s)$ randomly removes $|L^t| - s$ number of examples from L^t where s is computed according to Eq.10.

It is noteworthy that the initial classifiers in tri-training should be diverse because if all the classifiers are identical, then for any of these classifier, the unlabeled examples labeled by the other two classifiers will be the same as these labeled by the classifier for itself. Thus, tri-training degenerates to self-training with a single classifier. In the standard co-training algorithm, the use of sufficient and redundant views enables the classifiers be different. In fact, previous research has shown that even when there is no natural attributes partitions, if there are sufficient redundancy among the attributes then a fairly reasonable attribute partition will enable co-training exhibit advantages. While in the extended co-training algorithm which does not require sufficient and redundant views, the diversity among the classifiers is achieved through using different supervised learning algorithms. Since the tri-training algorithm does not assume sufficient and redundant views and different supervised learning algorithms, the diversity of the classifiers have to be sought from other channels. Indeed, here the diversity is obtained through manipulating the original labeled example set. In detail, the initial classifiers are trained from data sets generated via bootstrap sampling from the original labeled example set. These classifiers are then refined in the tri-training process, and the final hypothesis is produced via majority voting. The generation of the initial classifiers looks

like training an ensemble from the labeled example set with a popular ensembel learning algorithm, that is, Bagging.

Tri-training can be regarded as a new extension to the co-training algorithms. As mentioned before, Blum and Mitchell's algorithm requires the instance space be described by two sufficient and redundant views, which can hardly be satisfied in common data mining scenarios. Since tri-training does not rely on different views, its applicability is broader. Goldman and Zhou's algorithm does not rely on different views either. However, their algorithm requires two different supervised learning algorithms that partition the instance space into a set of equivalence classes. Moreover, their algorithm frequently uses 10-fold cross validation on the original labeled example set to determine how to label the unlabeled examples and how to produce the final hypothesis. If the original labeled example set is rather small, cross validation will exhibit high variance and is not helpful for model selection. Also, the frequently used cross validation makes the learning process time-consuming. Since tri-training does not put any constraint on the supervised learning algorithm nor does it employ time-consuming cross validation processes, both its applicability and efficiency are better.

$(t^{'}-^{'}1)$