Learning with Labeled Sessions

Rong Jin*, Huan Liu†
*Dept. of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824
rongjin@cse.msu.edu
† Department of Computer Science and Engineering, Arizona State University, Tempe, AZ85287-8809
hliu@asu.edu

Abstract

Traditional supervised learning deals with labeled instances. In many applications such as physiological data modeling and speaker identification, however, training examples are often *labeled objects* and each of the labeled objects consists of *multiple unlabeled instances*. When classifying a new object, its class is determined by the majority of its instance classes. As a consequence of this decision rule, one challenge to learning with labeled objects (or sessions) is to determine during training which subset of the instances inside an object should belong to the class of the object. We call this type of learning '*session-based learning*' to distinguish it from the traditional supervised learning. In this paper, we introduce session-based learning problems, give a formal description of session-based learning in the context of related work, and propose an approach that is particularly designed for session-based learning. Empirical studies with UCI datasets and real-world data show that the proposed approach is effective for session-based learning.

1 Introduction

A typical supervised learning problem is to learn a model from training examples that are usually labeled instances. But for many applications, training examples are *labeled* objects and each labeled object consists of *multiple unlabeled* instances. During classification, an object is labeled as class '*a*' if the majority of its instances are classified as '*a*'.

One application is physiological data modeling, whose goal is to predict context activities of individual users based on their physiological data. Typically, physiological signals are measured and recorded continuously for every other second. Continuous physiological signals are then divided into a number of sessions. Each session is of minutes long and consists of hundreds of records of physiological data. To predict the user's activities for a session, prediction is first made for each record, and the dominant activity predicted for records in the session is then used as the activity for the session. Although each session is labeled as a single activity, a user may perform activities other than the labeled one. For example, during a session of 'watching TV', the user may fall into sleep for a short period of time. More information can be found from the website http://www.cs.utexas.edu/users/sherstov/pdmc/.

Another application is speaker identification. To determine the speaker for a speech sample that is a minute long, a typical strategy is to first divide the long speech sample into a number of short ones. Then, a standard classification model, such as Gaussian Mixture Model (GMM), is applied to determine the speaker identity for each short sample. Finally, the dominant speaker that is classified for short samples will be used as the predicted speaker for the long sample. Compared with the strategy that extracts a single set of features for the whole long speech sample, this majority vote approach is usually more robust and accurate [Reynolds, 1995]. This is because features extracted from a long speech sample may include significant amounts of background noise, while the noise can be reduced substantially when a long speech sample is divided into short ones.

The common characteristics of the above two applications are that (1) training examples are labeled objects (e.g., sessions of physiological records in physiological data modeling, and a long speech sample in speaker identification); (2) every object consists of multiple instances that are not labeled; and (3) in predicting an object's class, it is determined by the class that is assigned to the majority of its instances. To distinguish this new type of learning from the traditional supervised learning, we call it '**session-based learning**'.

The challenge of session-based learning arises from the majority vote strategy that is used to determine the label of an object. This decision rule makes it ambiguous, during training, as to which subset of the instances inside an object should belong to the class of the object – we name it the **label ambiguity problem**. A straightforward strategy toward this problem is to treat every unlabeled instance within a labeled object as a positive example for the class of the object. In physiological data modeling, every physiological record within a labeled session is treated as a training instance for the activity assigned to the session. In speaker identification, every short sample within a long speech is used as a positive training instance for the speaker of the long speech. We call this simple strategy the '**naive approach**'.

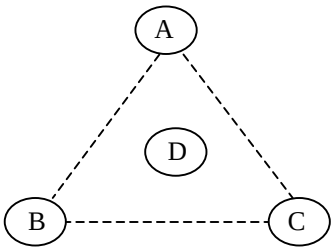

Figure 1: A toy example of session-based learning problem with four classes 'A', 'B', 'C', and 'D'.

One obvious problem with the naïve approach is that instances within a single object often belong to multiple different classes, not just to the class of the object. Thus, by treating every instance within an object as a positive example for the class of the object, we likely introduce training examples with noisy labels, which, as a result, degrades the quality of classification models. To further illustrate the problem with the naïve approach, let us consider a toy learning problem in Figure 1. It has four classes, with class 'A', 'B', and 'C' centering on the vertices of a triangle, and the fourth class 'D' sitting on the center of the triangle. In traditional supervised learning, training examples are labeled instances. Since these four classes are well separated, we would expect that a simple Naïve Bayes model should work fine for this problem. But, for a session-based learning problem, training examples are labeled objects that consist of instances from different classes. In particular, consider the case when training objects for classes 'A', 'B', and 'C' are mixtures of instances from these three classes, while training objects for class 'D' only contain instances from 'D'. With appropriate mixtures, the input means of instances from labeled objects for the four classes can stay close to each other, which makes it hard for the Naïve Bayes method to learn if assigning each instance with its object class.

In the following, we first give a formal description of session-based learning, and elaborate on the differences between this new type of learning problem and other related learning problems, such as multiple-instance learning. Then, we present a novel approach that is particularly designed for session-based learning. The key idea is to develop an innovative way that handles the label ambiguity problem. Finally, we demonstrate the effectiveness of the proposed approach for session-based learning using both UCI datasets and data from physiological data modeling.

2 Formal Description of Session-based Learning

Let training examples be denoted by $D = \{(o_1, y_1), (o_2, y_2), ..., (o_n, y_n)\}$, where each (o_i, y_i) is a labeled object and $y_i \in \mathbf{Y}$ is the class label assigned to object o_i. Domain $\mathbf{Y}$ is $\{-1,1\}$ for binary-class classification problems, and $[1..K]$ for multiple-class classification problems where K ($K > 2$) is the number of classes. Let $c(o)$ be an object-based classification function, which takes an object o as input and outputs its class label. In general, any supervised learning problem can be formulated as an optimization problem:

$$c^* = \arg\min_c \sum_{i=1}^{n} L\left(c(o_i), y_i\right) \qquad (1)$$

where L is a loss function that determines the amount of punishment when prediction $c(o_i)$ is different from y_i.

For a traditional supervised learning problem, each object only contains a single instance. As a result, training examples can be simplified as $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)\}$, where each instance $\mathbf{x}_i \in \Re^d$ is a vector in a d dimension space. Furthermore, the object-based classification function $c(o)$ becomes an instance-based classification function $f(\mathbf{x}): \Re^d \to \mathbf{Y}$. Thus, a traditional supervised learning problem is usually formulated as follows:

$$f^* = \arg\min_f \sum_{i=1}^{n} L\left(f(\mathbf{x}_i), y_i\right) \qquad (1a)$$

In session-based learning, each object consists of multiple unlabeled instances. Let the training data be denoted by $D = \{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), ..., (\mathbf{X}_n, y_n)\}$, where each object $\mathbf{X}_i = \{\mathbf{x}_{i,j}\}_{j=1}^{m_i}$ contains m_i different instances. Given an instance-based classification function $f(\mathbf{x}): \Re^d \to \mathbf{Y}$, the decision rule for session-based learning is that, an object $\mathbf{X}_i$ is labeled as class '*a*' if the majority of its instances are classified as '*a*'. Thus, in session-based learning, the object-based classification function c_s can be written into the following form of the instance-based classification $f(\mathbf{x})$:

$$c_s(\mathbf{X}_i; f) = \arg\max_{y \in \mathbf{Y}} \sum_{j=1}^{m_i} \delta\left(f(\mathbf{x}_{i,j}) = y\right) \qquad (2)$$

where $\delta(\cdot)$ is a delta function that outputs 1 when the input is positive and zero otherwise.

Compared to traditional supervised learning, the challenge of session-based learning is due to the label ambiguity problem. Although each training object is provided with a class label, the label information of instances within objects is not given. Hence, it is difficult to learn an instance-based classification function from labeled objects. In the naïve approach, each instance within an object is treated as a positive example for the class of the object. As a result, a session-based learning problem is simplified as a traditional supervised learning problem:

$$f^* = \arg\min_f \sum_{i=1}^{n} \sum_{j=1}^{m_i} L\left(f(\mathbf{x}_{i,j}), y_i\right) \qquad (1b)$$

The most related work to session-based learning is multiple-instance learning[Dietterich, et al.,1997]. Similar to session-based learning, in multiple-instance learning, class la-

bels are assigned to objects that consist of multiple instances, which are called "bags" in multiple-instance learning. In the past, there have been many studies on multiple-instance learning, including the approach of learning axis-parallel rectangles [Dietterich, et al.,1997], the diverse density algorithm [Maron and Lozano-Pérez,1998], the approaches based on support vector machines [Andrews, et al.,2002, Tao, et al.,2004], the nearest neighbor approach [Amar et.al. 2001; Wang and Zucker, 2000], and the boosting approach [Andrews and Hofmann,2003].

Multiple-instance learning differs from session-based learning in its decision rule. In multiple-instance learning, an object $\mathbf{X}_i$ is labeled as positive class when at least one of its instances is classified as positive. A negative class is assigned to an object when all of its instances are classified as negative. Thus, given the instance-based classification function $f(\mathbf{x}):\Re^d \to \mathbf{Y}$, the object-based classification function c_m for multiple-instance learning is written as:

$$c_m(\mathbf{X}_i; f) = \begin{cases} +1 & \exists j \in [1...m_i],\ f(\mathbf{x}_{i,j}) = +1 \\ -1 & \forall i \in [1...m_i],\ f(\mathbf{x}_{i,j}) = -1 \end{cases} \qquad (2')$$

Examples for the three different types of learning are shown in Figure 2 for comparison. The first column shows traditional supervised learning with instances and their labels. The columns for Multiple Instance Learning and Session-based Learning show how these two different strategies aggregate instances and assign labels to the aggregates.

Session-based learning is more challenging than multiple-instance learning in the following sense: in multiple-instance learning, when an object is labeled as 'negative', all of its instances will belong to the negative class. Thus, for multiple-instance learning, there is no label ambiguity for negatively labeled objects. In contrast, the label ambiguity problem exists for session-based learning regardless of the sign of labels. For instance, in Figure 2, for session-based learning, both the first and second objects are labeled as 'negative'. However, the labels of instances in these two objects are different. The two learning schemes also differ in degrees of difficulty when deciding positive classes for objects: multiple-instance learning adopts the "at-least one" strategy and session based learning uses the majority one.

3 SBoost – An Algorithm for Session-based Learning

In this session, we will present a boosting-based algorithm for session-based learning problems of binary classes. The key for designing a learning algorithm for session-based learning is to define a simple yet effective loss function $L(c(\mathbf{X}_i; f), y_i)$. A simple choice is $L(c(\mathbf{X}_i; f), y_i) = \delta(c(\mathbf{X}_i; f) \neq y_i)$. However, this choice will lead to a non-smooth objective function, which is usually difficult for optimization. Hence, we choose to use the exponential loss function to approximate classification errors, which has been demonstrated to be effective in the AdaBoost algorithm [Freund and Schapire, 1997].

Trad. Supervised Learning		Multiple-instance Learning		Session-based Learning	
0.1 0.2 0.3	-1	0.1 0.2 0.3	-1	0.1 0.2 0.3	-1
0.2 0.5 0.4	-1	0.2 0.5 0.4		0.2 0.5 0.4	
0.6 0.2 0.1	-1	0.6 0.2 0.1		0.6 0.2 0.1	
……		……		……	
0.3 0.7 0.6	-1	0.3 0.7 0.6	+1	0.3 0.7 0.6	-1
0.7 0.2 0.2	-1	0.7 0.2 0.2		0.7 0.2 0.2	
0.8 0.1 0.0	+1	0.8 0.1 0.0		0.8 0.1 0.0	
……		……		……	
0.5 0.7 0.1	+1	0.5 0.7 0.1	+1	0.5 0.7 0.1	+1
0.1 0.6 0.9	-1	0.1 0.6 0.9		0.1 0.6 0.9	
0.7 0.1 0.2	+1	0.7 0.1 0.2		0.7 0.1 0.2	

Figure 2: Training examples for traditional supervised learning, multiple-instance learning, and session-based leaning.

In particular, in designing objective functions, two types of errors are considered: instance-based errors and session-based errors. An instance-based error is a prediction mistake made for an instance, and a session-based error is a prediction mistake made for a session. One key difference between session-based learning and traditional supervised learning is that the former is concerned with both session-based errors and instance-based errors while the latter concerns with only instance-based errors. To include both types of errors, given an instance-based classification function $H(x)$, we define the loss function as:

$$L(c(X_i; H), y_i) = \underbrace{\exp\left(-\frac{\gamma y_i}{m_i}\sum_{j=1}^{m_i} H(x_{i,j})\right)}_{\text{Session-based error}} \underbrace{\left\{\sum_{j=1}^{m_i}\exp\left(-H(x_{i,j})y_i\right)\right\}}_{\text{Instance-based error}} \qquad (3)$$

In the above expression, both session-based and instance-based errors are approximated by an exponential function. The loss function is defined as the product of these two errors. In other words, a misclassified instance is important only when its related session is also misclassified. Constant γ in (3) determines the relative importance between the session-based error and the instance-based error. In experiment, a cross validation with 20/80 split of training data is used to determine appropriate values for γ. In the following, we will discuss how to efficiently find $H(x)$ that minimizes the loss function in (3).

3.1 Boosting-based Optimization Algorithm

With the loss function defined in (3), our goal is then to search for optimal $H(x)$ that minimizes the overall cost for the training data, i.e.,

$$H^* = \arg\min_H err = \arg\min_H \sum_{i=1}^{n} L(c(\mathbf{X}_i; H), y_i)$$
$$= \arg\min_H \sum_{i=1}^{n} \exp\left(-\frac{\gamma y_i}{m_i}\sum_{j=1}^{m_i} H(x_{i,j})\right)\left\{\sum_{j=1}^{m_i}\exp\left(-H(x_{i,j})y_i\right)\right\} \qquad (4)$$

Given: $(\mathbf{X}_1, y_1), \ldots, (\mathbf{X}_m, y_m)$ where $\mathbf{X}_i = \left\{\mathbf{x}_{i,j}\right\}_{j=1}^{m_i}$, $\mathbf{x}_{i,j} \in \Re^d$, $y_i \in \{-1,1\}$; Weighting constant γ

Initialize the weight distribution $D_0(i,j) = 1 \Big/ \left(\sum_{i=1}^{n} m_i\right)$, $H_0(x) = 0$

For *t* = 1,…,*T*

1. Sample training instances $\left\{\mathbf{x}_{1,1},...,\mathbf{x}_{1,m_1},...,\mathbf{x}_{n,1},...,\mathbf{x}_{n,m_n}\right\}$ according to D_{t-1}
2. Train a weak classifier $h_t(\mathbf{x}): \Re^d \to \{-1,1\}$ on sampled examples
3. Compute $g_i = \exp\left(-\frac{y_i \gamma}{m_i} \sum_{j=1}^{m_i} H_{t-1}(\mathbf{x}_{i,j})\right)$, $a_i = \sum_{j=1}^{m_i} \exp\left(-H_{t-1}(\mathbf{x}_{i,j}) y_i\right)$, and
   $b_i = \sum_{j=1}^{m_i} y_i h(\mathbf{x}_{i,j}) \exp\left(-H_{t-1}(\mathbf{x}_{i,j}) y_i\right) + \frac{y_i a_i \gamma}{m_i} \sum_{j=1}^{m_i} h(\mathbf{x}_{i,j})$ for each session.
4. Let $\alpha_t = \frac{1}{2(1+\gamma)} \ln\left\{\frac{\sum_{i=1}^{n} g_i \left[(1+\gamma)a_i + b_i\right]}{\sum_{i=1}^{n} g_i \left[(1+\gamma)a_i - b_i\right]}\right\}$
5. Update the weight distribution $D_t(i,j) = \frac{g_i \left(\exp\left(-H_{t-1}(\mathbf{x}_{i,j}) y_i\right) + \gamma a_i / m_i\right)}{Z_t}$, where Z_t is a normalization factor (chosen so that D_{t+1} is sum to 1).
6. Update the classifier $H_t(\mathbf{x}) = H_{t-1}(\mathbf{x}) + \alpha_t h_t(\mathbf{x})$

Output the final hypothesis: $F_T(\mathbf{x}) = \arg\max_{y \in \{-1,1\}} \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) y$

Figure 3: Description of the SBoost algorithm.

An efficient approach for optimizing Eq. (4) is to divide it into a series of simple learning problems that do not have the label ambiguity problem and thus can be resolved by traditional supervised learning techniques. We solve the label ambiguity problem by maintaining weights for different instances such that only instances with large weights are assigned to the class of their objects and used for training. Since boosting [Freund and Schapire, 1997] is a learning algorithm that uses weighted instances to efficiently update classification functions, we design a boosting-based learning algorithm for learning with labeled sessions below.

Let $h_t(\mathbf{x})$ be the 'weak' classifier of the *t*-th iteration that is learned using traditional supervised learning techniques. The combined classifier $H_T(\mathbf{x})$ for the first *T* iterations is $H_T(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$, where α_t is the combination constant for the *t*-th iteration. Our goal is to find another 'weak' classifier $h_{T+1}(\mathbf{x})$ and a constant α_{T+1} such that the new combined classifier $H_{T+1}(\mathbf{x}) = H_T(\mathbf{x}) + \alpha_{T+1} h_{T+1}(\mathbf{x})$ will effectively minimize the function in (4). Given classifier $H_{T+1}(\mathbf{x})$, the objective function in (4) is rewritten as:

$$err = \sum_{i=1}^{n} \left\{ \begin{array}{l} \exp\left(-\frac{\gamma y_i}{m_i} \sum_{j=1}^{m_i} \left[H(\mathbf{x}_{i,j}) + \alpha h(\mathbf{x}_{i,j})\right]\right) \\ \times \sum_{j=1}^{m_i} \exp\left(-\left[H(x_{i,j}) + \alpha h(x_{i,j})\right] y_i\right) \end{array} \right\} \quad (5)$$

In the above expression, for the convenience of presentation, we drop the index for $H_T(\mathbf{x})$, $h_{T+1}(\mathbf{x})$, and α_{T+1}. Using the convexity of an exponential function, we have

$$err \le \sum_{i=1}^{n} \frac{g_i}{m_i} \sum_{k,j=1}^{m_i} \exp\left(-\alpha\left[h(\mathbf{x}_{i,j}) + \gamma h(\mathbf{x}_{i,k})\right] y_i - H(\mathbf{x}_{i,j}) y_i\right) \quad (6)$$

where $g_i \equiv \exp\left(-\frac{y_i \gamma}{m_i} \sum_{j=1}^{m_i} H(\mathbf{x}_{i,j})\right)$. Then, using inequality $e^{\alpha x} \le \frac{1+x}{2} e^{\alpha} + \frac{1-x}{2} e^{-\alpha}$, $\forall x \in [-1,1]$, we can further upper bound (6) by the following expression:

$$err \le err_{\text{upper}}$$
$$= \frac{e^{\alpha(1+\gamma)} + e^{-\alpha(1+\gamma)}}{2} \sum_{i=1}^{n} a_i g_i$$
$$- \frac{e^{\alpha(1+\gamma)} - e^{-\alpha(1+\gamma)}}{2(1+\gamma)} \sum_{i=1}^{n} \sum_{j=1}^{m_i} y_i h(\mathbf{x}_{i,j}) \left\{ \begin{array}{l} g_i \exp\left(-H(\mathbf{x}_{i,j}) y_i\right) \\ + g_i \dfrac{\gamma a_i}{m_i} \end{array} \right\} \quad (7)$$

where $a_i \equiv \sum_{j=1}^{m_i} \exp\left(-H(\mathbf{x}_{i,j}) y_i\right)$. Define weight $D(i,j) \equiv g_i \left(\exp\left(-H(\mathbf{x}_{i,j}) y_i\right) + \gamma a_i / m_i\right)$ and rewrite the second term in (7) as:

$$\frac{e^{\alpha(1+\gamma)} - e^{-\alpha(1+\gamma)}}{2(1+\gamma)} \sum_{i=1}^{n} \sum_{j=1}^{m_i} y_i h(\mathbf{x}_{i,j}) D(i,j) \quad (8)$$

Clearly, to minimize the objective *err* in (7), we need to maximize the above expression (8). The best case is that the output of weak classifier $h(\mathbf{x})$ is consistent with y_i, for all instances in an object. When this is not the case, the label ambiguity problem is then resolved based on weight $D(i,j)$. In particular, instances with large weights are assigned with the class of their objects, while the labels for instances with small weights remain unlabeled. This is because the contribution of an instance to (8) is mainly determined by its weight $D(i,j)$. When an instance has a small weight, its contribution to (8) will be ignorable. Furthermore, notice that $D(i,j)$ is proportional to g_i, which is related to session-based error. Thus, a misclassified instance will not be assigned with a large weight if the related session error is small. Finally, the combination constant α can be obtained by setting the derivative of the upper bound in (7) w.r.t. to α to be zero. That is,

$$\alpha = \frac{1}{2(1+\gamma)} \ln \left\{ \frac{\sum_{i=1}^{n} g_i \left[(1+\gamma) a_i + b_i\right]}{\sum_{i=1}^{n} g_i \left[(1+\gamma) a_i - b_i\right]} \right\} \quad (9)$$

where

$$b_i \equiv \sum_{j=1}^{m_i} y_i h(x_{i,j}) \exp\left(-H(\mathbf{x}_{i,j}) y_i\right) + \frac{y_i a_i \gamma}{m_i} \sum_{j=1}^{m_i} h(\mathbf{x}_{i,j}).$$

For later reference, we name this algorithm **SBoost** for '**session-based boosting**'. The details are given in Figure 3.

4 Experiments

The goal of this section is to examine the effectiveness of SBoost for session-based learning. We compare SBoost with the simple naïve approach that treats each instance within an object as a positive example for the class of the object. In particular, the naïve approach is applied to session-based learning with both a Decision Tree and AdaBoost using a Decision Tree as its base classifier.

Data Set	# Examples	# Features
spam	4600	58
cmc	1470	10
german	680	24

Table 1: Statistics of UCI datasets used for synthesized data

	Positive Class	Negative Class
# Instances	4343	3,6495
# Objects	55	186
# Avg of instances per object	69.3455	199.0538

Table 2: Statistics for the physiological data

Two types of data are used in experiments to evaluate the effectiveness of SBoost:

1) ***Synthesized data*** that are generated from binary UCI datasets [Blake and Merz,1998] by combining multiple instances into objects. Each object consists of ten different instances. To create an object for the positive class, a random number between 1 and 5 is first generated, and the corresponding number of instances from the negative class are randomly chosen and added to the object. The rest of the object is filled out with instances randomly selected from the positive class. A similar procedure is applied to generate objects for the negative class. By doing so, we guarantee that the class of an object is consistent with the dominant class assigned to its instances. The details of UCI datasets used in this experiment are listed in Table 1.

2) ***Physiological data*** that come from the workshop of physiological data modeling at the ICML 2004. We use the dataset for 'watching TV' with code 3004. In the original problem, the number of instances for the negative class is overwhelmingly larger than that for the positive class. Since we focus on the study of session-based learning, we intentionally reduce the effect of rare class by randomly selecting parts of negative instances. The resulting data set has 241 sessions with 40,838 instances. The details of this dataset are listed in Table 2.

A decision tree [Quinlan,1993] is used as the baseline classifier throughout the experiments. The session-based classification error, i.e., the percentage of sessions that are misclassified, is used for evaluation. 50% of data are randomly selected for training and the rest is used for testing. The same experiment is repeated 10 times, and the average session-based classification errors are reported. Finally, for both SBoost and AdaBoost, the maximum number of iterations is set to be 30.

4.1 Results

The results of three methods (SBoost, the naïve approach using a single decision tree, and with AdaBoost) are shown in Table 3. First, we compare the performance of AdaBoost with that of a decision tree as both adopting the naïve approach to session based learning. We observe that AdaBoost

Data Set	SBoost	AdaBoost	Decision Tree
spam	0.189 (0.022)	0.258 (0.059)	0.232 (0.051)
cmc	0.439 (0.044)	0.444 (0.070)	0.519 (0.052)
german	0.342 (0.028)	0.424 (0.069)	0.476 (0.081)
phys.	0.163 (0.038)	0.191 (0.035)	0.176(0.030)

Table 3: Classification errors with variances for the SBoost algorithm, the AdaBoost, and the decision tree

does not guarantee to improve performance over the baseline classifier (a decision tree here). In fact, for the spam dataset and the physiological data, the classification errors of AdaBoost are even greater than those of a decision tree. This is because the naïve approach treats all instances in an object as positive examples for the class of the object, while in reality, some instances in an object may belong to a class other than the class of the object. Hence, the naïve approach introduces noisy labels to training data, which will likely cause AdaBoost to overfit as observed in previous study of Boosting [Quanlin, 1996, Dietterich, 2000, Jin et al., 2003].

Second, comparing SBoost with the decision tree, we observe that SBoost always outperforms its baseline classifier. According to t-test, it is statistically significantly better than the decision over all the UCI datasets with $p < 0.05$. Thus, SBoost does not suffer from the problem of overfitting as AdaBoost does. To further investigate this issue, we obtain average classification errors of AdaBoost and SBoost in different iterations, respectively, as shown in Figure 4. Clearly, AdaBoost tends to overfit the training data after the first several iterations whereas SBoost does not.

5. Conclusion

In this paper, we formulate a new type of learning problem, session-based learning. It differs from the traditional supervised learning in that training examples are labeled objects and each object consists of multiple unlabeled instances. Furthermore, session-based learning adopts a different decision rule from that of multiple-instance learning: an object is classified as class '*a*' when the majority of its instances are classified as '*a*'. This majority vote decision rule causes the label ambiguity problem and has made session-based learning very challenging. We formally describe this new learning problem and propose a novel boosting algorithm, named Sboost. Empirical studies have demonstrated the effectiveness of the Sboost algorithm.

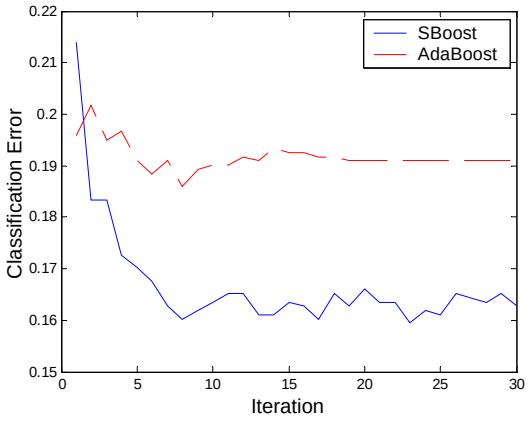

Figure 4: Classification errors for physiological data

References

[Amar et.al. 2001] Robert A. Amar, Daniel R. Dooly, Sally A. Goldman, Qi Zhang, *Multiple-Instance Learning of Real-Valued Data*, Proc. 18th ICML, 2001.

[Andrews and Hofmann, 2003] Stuart Andrews and Thomas Hofmann, *Multiple-Instance Learning via Disjunctive Programming Boosting*, NIPS, 2003.

[Andrews, et al., 2002] Stuart Andrews, Thomas Hofmann and Ioannis Tsochantaridis, *Multiple Instance Learning with Generalized Support Vector Machines*, Proc. 18th AAAI , 2002.

[Blake and Merz,1998] C.L. Blake and C.J. Merz, *UCI Repository of Machine Learning Databases*,1998.

[Dietterich, et al.,1997] T. G. Dietterich, Richard H. Lathrop and Tomas Lozano-Perez, *Solving the Multiple-Instance Problem with Axis-Parallel Rectangles*, Artificial Intelligence, 89(1-2), 31-71,1997.

[Dietterich, 2000] T. G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning, 40:139–157,2000.

[Freund and Schapire, 1997] Yoav Freund and Robert E. Schapire, *A Decision-theoretic Generalization of On-line Learning and an Application to Boosting*, Journal of Computer and System Sciences, 55(1), 119-139, 1997.

[Maron and Lozano-Pérez,1998] Oded Maron and Tomás Lozano-Pérez, *A Framework for Multiple-Instance Learning*, NIPS, 1998.

[Jin et al. ,2003] Rong Jin, Yan Liu , Luo Si, Jaime Carbonell, Alexander G. Hauptmann, *A New Boosting Algorithm Using Input-Dependent Regularizer*, Proc. 20th ICML, 2003

[Quinlan,1993] R.J. Quinlan,C4.5: *Programs for Machine Learning*, Morgan Kaufmann,1993.

[Tao, et al.,2004] Qingping Tao, Stephen Scott, N. V. Vinodchandran and Thomas Takeo Osugi, *SVM-based Generalized Multiple-Instance Learning via Approximate Box Counting*, Proc. 21st ICML, 2004.

[Wang and Zucker, 2000] Jun Wang, Jean-Daniel Zucker, *Solving the Multiple-Instance Problem: A Lazy Learning Approach*, Proc. 17th ICML, 2000.

[Quinlan, 1996] Quinlan, J. R. (1996). *Bagging, Boosting, and C4.5*. Proceedings of the 13th National Conference on Artificial Intelligence, 322–330.

[Reynolds, 1995] D. A. Reynolds, *Speaker identification and verification using Gaussian mixture speaker models*, Speech Communications, vol. 17, pp. 91–108, 1995.