Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, July 31 - August 4, 2005

A Study of AdaBoost with SVM Based Weak Learners

Xuchun Li, Lei Wang, Eric Sung
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore, 639798
E-mail: xuchunli@pmail.ntu.edu.sg, elwang@ntu.edu.sg, eericsung@ntu.edu.sg

Abstract— **In this article, we focus on designing an algorithm, named *AdaBoostSVM*, using SVM as weak learners for AdaBoost. To obtain a set of effective SVM weak learners, this algorithm adaptively adjusts the kernel parameter in SVM instead of using a fixed one. Compared with the existing AdaBoost methods, the *AdaBoostSVM* has advantages of easier model selection and better generalization performance. It also provides a possible way to handle the over-fitting problem in AdaBoost. An improved version called *Diverse AdaBoostSVM* is further developed to deal with the accuracy/diveristy dilemma in Boosting methods. By implementing some parameter adjusting strategies, the distributions of accuracy and diversity over these SVM weak learners are tuned to achieve a good balance. To the best of our knowledge, such a mechanism that can conveniently and explicitly balances this dilemma has not been seen in the literature. Experimental results demonstrated that both proposed algorithms achieve better generalization performance than AdaBoost using other kinds of weak learners. Benefiting from the balance between accuracy and diversity, the *Diverse AdaBoostSVM* achieves the best performance. In addition, the experiments on unbalanced data sets showed that the *AdaBoostSVM* performed much better than SVM.**

I. INTRODUCTION

One of the major developments in machine learning in the past decade is the ensemble method, which finds a highly accurate classifier by combining many moderately accurate component classifiers. Two of the commonly used techniques for constructing ensemble classifiers are Boosting [1] and Bagging [2]. Comparing with Bagging, Boosting performs better in most situations [3]. As the most popular Boosting method, AdaBoost [4] creates a collection of weak learners by maintaining a set of weights over training samples and adjusting these weights after each weak learning cycle adaptively: the weights of the samples which are misclassified by current weak learner will be increased while the weights of the samples which are correctly classified will be decreased.

The success of AdaBoost can be explained as enlarging the margin [5], which could enhance AdaBoost's generalization capability. Many studies that use decision trees [6] or neural networks [7] [8] as weak learners for AdaBoost have been reported. These studies showed the good generalization performance of AdaBoost. However, when decision trees are used as weak learners, what is the suitable tree size? When RBF neural networks are used as weak learners, how to control their complexity to avoid overfitting? How to decide the number of centers and how to set the width values of the RBFs? All of these have to be carefully tuned in practical use of AdaBoost. Furthermore, *diversity* is known as an important factor which affects the generalization accuracy of ensemble classifiers [9][10]. It is also known that AdaBoost exists an accuracy/diversity dilemma [6], which means that the more accurate two weak learners become, the less they can disagree with each other. However, the existing AdaBoost algorithms have not yet explicitly taken sufficient measurement to deal with this dilemma. Finally, It is reported that AdaBoost may overfit the training samples [11] and result in poor generalization performance. Therefore, it is necessary to stop AdaBoost's learning cycles at a suitable moment. But how to truncate the AdaBoost learning process to avoid overfitting is still an open problem [11].

Support Vector Machine [12] was developed from the theory of Structural Risk Minimization. By using a kernel trick to map the training samples from an input space to a high dimensional feature space, SVM finds an optimal separating hyperplane in the feature space and uses a regularization parameter, C, to balance its model complexity and training error. One of the popular kernels used by SVM is the RBF kernel, which has a parameter known as Gaussian width, σ. Comparing with RBF networks, SVM with RBF kernel (RBFSVM) can automatically calculate the number and location of the centers and the weights [13]. Also, it can effectively avoid overfitting by selecting proper parameters C and σ. From the performance analysis of RBFSVM [14], we know that σ is a more important parameter: although RBFSVM cannot learn well when a very low value of C is used, its performance largely depends on the σ value if a roughly suitable C is given. This means that, for a given roughly suitable C, the performance of RBFSVM can be conveniently changed by simply adjusting the value of σ.

Therefore, in this paper, we try to answer the following questions: Can SVM be used as a weak learner of AdaBoost? If yes, how about the generalization performance of this AdaBoost? Does this AdaBoost have some advantages over the existing ones, especially about the aforementioned problems? Also, compared with using a single SVM, what is the benefit of using this AdaBoost which is a combination of multiple SVMs? In this work, the RBFSVM is adopted as a weak learner for AdaBoost. As mentioned above, the RBFSVM has a parameter of σ which has to be set beforehand. An intuitive way is to simply apply a single σ to all SVM weak

0-7803-9048-2/05/$20.00 ©2005 IEEE

learners. However, we observed that this way cannot lead to successful AdaBoost due to over-weak or over-strong SVM weak learners. Although there may exist a single best σ, searching for it will largely increase the computational load of Boosting process and therefore should be avoided if possible.

The following fact opens a door for us to avoid searching for the single best σ. It is known that the classification performance of RBFSVM can be conveniently changed by adjusting the value of σ. Based on this, the algorithm, *AdaBoostSVM*, is developed, where a set of moderately accurate RBFSVM is trained for AdaBoost by adaptively adjusting the σ values instead of using a fixed one. This gives rise to a successful SVM based AdaBoost. Compared with the existing AdaBoost methods, proposed *AdaBoostSVM* has the following advantages: It needs not perform accurate parameter tuning. Instead, giving a rough range of σ is often enough. By setting the lower end of this range, proposed algorithm also provides a possible way to truncate the Boosting process to handle the over-fitting problem when other kinds of weak learners are used. Besides these, proposed algorithm achieves better classification performance than AdaBoost with other weak learners such as neural networks.

Furthermore, since proposed *AdaBoostSVM* invents a convenient way to control the classification accuracy of each weak learner, it also provides an opportunity to deal with the well-known accuracy/diveristy dilemma in Boosting methods. This is a happy accident from the investigation of AdaBoost based on SVM weak learners. Through some parameter adjusting strategies, we can tune the distributions of accuracy and diversity over these weak learners to achieve a good balance. To the best of our knowledge, there is no such a mechanism which can conveniently and explicitly balance this dilemma in the literature. The improved version of *AdaBoostSVM* is called *Diverse AdaBoostSVM* in this work. It is observed that, benefiting from the balance between accuracy and diversity, it gives better performance than *AdaBoostSVM*.

Finally, compared with a single SVM, proposed algorithms, which are the combination of multiple SVMs, can achieve much better performance on unbalanced data sets. This also justifies, from another perspective, that AdaBoost with SVM weak learners is worth of investigation.

II. BACKGROUND

A. AdaBoost

Given a set of training samples, AdaBoost [15] maintains a probability distribution, W, over these samples. This distribution is initially uniform. Then, AdaBoost algorithm calls WeakLearn algorithm repeatedly in a series of cycles. At cycle t, AdaBoost provides training samples with a distribution W_t to the WeakLearn algorithm. In response, the WeakLearn trains a classifier h_t. The distribution W_t is updated after each cycle according to the prediction results on the training samples. "Easy" samples that are correctly classified by the weak learner, h_t, get low weights, and "hard" samples that are misclassified get higher weights. Thus, AdaBoost focuses on the samples with more weights, which seem to be harder for WeakLearn. This process continues for T cycles, and at last, AdaBoost uses weighted vote to combine all the obtained weak learners into a single final hypothesis f. Greater weights are given to weak learners with lower errors. The important theoretical property of AdaBoost is that if the weak learners consistently have accuracy only slightly better than half, then the error of the final hypothesis drops to zero exponentially fast. This means that the weak learners need be only slightly better than random.

TABLE I

```
Algorithm: AdaBoost [15]
1. Input: a set of training samples with labels $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$,
WeakLearn algorithm, the number of cycles $T$.
2. Initialize: the weight of samples: $w_i^1 = 1/N$, for all $i = 1, ..., N$.
3. Do for $t = 1, ..., T$
    (1)Use WeakLearn algorithm to train the weak learner $h_t$ on the
weighted training sample set.
    (2)Calculate the training error of $h_t : \epsilon_t = \sum_{i=1}^{N} w_i^t,\ y_i \neq h_t(\mathbf{x}_i)$.
    (3)Set weight of weak learner $h_t : \alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$.
    (4)Update training samples' weights: $w_i^{t+1} = \frac{w_i^t \exp\{-\alpha_t y_i h_t(\mathbf{x}_i)\}}{C_t}$
where $C_t$ is a normalization constant, and $\sum_{i=1}^{N} w_i^{t+1} = 1$.
4. Output: $f(\mathbf{x}) = sign(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}))$.
```

B. Support Vector Machine

Support Vector Machine (SVM) [12] was developed from the theory of Structural Risk Minimization. In a binary classification problem, SVM's decision function is

$$f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b \qquad (1)$$

where $\phi(\mathbf{x})$ is a mapping of sample $\mathbf{x}$ from the input space to a high-dimensional feature space. $\langle \cdot, \cdot \rangle$ denotes the dot product in the feature space. The optimal values of $\mathbf{w}$ and b can be obtained by solving the following optimization problem,

$$\textit{minimize:} \quad g(\mathbf{w}, \xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i \qquad (2)$$

$$\textit{subject to:} \quad y_i(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i,\ \xi_i \geq 0 \qquad (3)$$

Here, ξ_i is the i-th slack variable and C is the regularization parameter. According to the Wolfe dual form, the above minimization problem can be written as

$$\textit{mininize:} \quad W(\alpha) = -\sum_{i=1}^{N} \alpha_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \qquad (4)$$

$$\textit{subject to:} \quad \sum_{i=1}^{N} y_i \alpha_i = 0, \quad \forall i : 0 \leq \alpha_i \leq C \qquad (5)$$

where α_i is a Lagrange multiplier which corresponds to the sample $\mathbf{x}_i$, $k(\cdot, \cdot)$ is a kernel function that implicitly maps the input vectors into a suitable feature space

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \qquad (6)$$

Compared with the RBF networks [13], SVM algorithm automatically computes the number and location of the centers, the weights, and the threshold in the following way: by the use of a suitable kernel function(in this paper, the RBF kernel is used), the samples are mapped nonlinearly into a high dimensional feature space. In this space, an optimal separating hyperplane is constructed by the support vectors which are closest to the decision boundary. Support vectors correspond to the centers of RBF kernels in the input space.

III. PROPOSED ALGORITHM: ADABOOSTSVM

This work uses the RBFSVM as weak learner for AdaBoost. But how to set the σ value for these weak learners? Problems are encountered when applying a single σ to all weak learners. In detail, an over-large σ often results in too weak RBFSVM. Its classification accuracy is often less than 50% and cannot meet the requirement on a weak learner given in AdaBoost. On the other hand, a smaller σ often makes the RBFSVM stronger and boosting them may become inefficient because the errors of these weak learners are highly correlated. Furthermore, too small σ can even make RBFSVM overfit the training samples and they also cannot be used as weak learners. Hence, finding a suitable σ for AdaBoost with SVM weak learners becomes a problem. By using model selection techniques such as cross-validation or leave-one-out, a single best σ may be found, with which the AdaBoost may achieve good classification performance. However, the process of model selection is time-consuming and should be avoided if possible.

The classification performance of SVM is affected by its parameters. For RBFSVM, they are the Gaussian width, σ, and the regularization parameter, C. The variation of either of them leads to the change of classification performance. However, as reported in [14], although RBFSVM cannot learn well when a very low value of C is used, its performance largely depends on the σ value if a roughly suitable C is given. This means that, for a given C, the performance of RBFSVM can be changed by simply adjusting the value of σ. Increasing this value often reduces the learning model complexity and then lowers down the classification performance, whereas decreasing it can lead to more complex learning model and higher performance. Therefore, this gives a chance to get around the problem resulting from using a fixed σ for all RBFSVM weak learners. In the following proposed algorithm, we generate a set of moderately accurate RBFSVM classifiers for AdaBoost by adaptively adjusting the σ values.

When applying Boosting method to strong component classifiers, these classifiers must be appropriately weakened in order to benefit from Boosting [6]. Hence, if RBFSVM is used as weak learner for AdaBoost, a relatively large σ value, which corresponds to a RBFSVM with relatively weak learning ability, is preferred. Both re-sampling and re-weighting can be used to train AdaBoost. In the proposed algorithm, without loss of generality, re-weighting technique is used. We proposed an *AdaBoostSVM* algorithm as follows (Table. II): Firstly, a large value is set to σ, which corresponds to a RBFSVM classifier with very weak learning ability. Then, RBFSVM with this σ is used in as many cycles as possible as long as more than half accuracy on weighted training samples can be obtained. Otherwise, this σ value is decreased slightly to make the RBFSVM with the new σ value obtain more than half accuracy. This process continues until the σ is decreased to the given minimal σ value. By doing so, the proposed AdaBoostSVM algorithm can often generate a set of moderately accurate SVM classifiers with possibly uncorrelated errors.

TABLE II

```
Algorithm: AdaBoostSVM
1. Input: a set of training samples with labels $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$;
the initial $\sigma$, $\sigma_{ini}$; the minimal $\sigma$, $\sigma_{min}$; the step of $\sigma$, $\sigma_{step}$.
2. Initialize: the weight of samples: $w_i^1 = 1/N$, for all $i = 1, ..., N$.
3. Do While $(\sigma > \sigma_{min})$
  (1)Use RBFSVM algorithm to train the weak learner $h_t$ on the weighted training sample set.
  (2)Calculate training error of $h_t : \epsilon_t = \sum_{i=1}^{N} w_i^t, y_i \neq h_t(\mathbf{x}_i)$.
  (3)If $\epsilon_t > 0.5$, decrease $\sigma$ value by $\sigma_{step}$ and goto (1).
  (4)Set weight of weak learner $h_t : \alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$.
  (5)Update training samples' weights: $w_i^{t+1} = \frac{w_i^t \exp\{-\alpha_t y_i h_t(\mathbf{x}_i)\}}{C_t}$
where $C_t$ is a normalization constant, and $\sum_{i=1}^{N} w_i^{t+1} = 1$.
4. Output: $f(\mathbf{x}) = sign(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}))$.
```

IV. IMPROVEMENT: DIVERSE ADABOOSTSVM

A. Accuracy/Diversity Dilemma of AdaBoost

Diversity is known to be an important factor affecting the generalization performance of ensemble methods [9][10], which means that the errors made by different component classifiers are uncorrelated. If each component classifier is moderately accurate and these component classifiers disagree with each other, the uncorrelated errors of these component classifiers will be removed by the voting process so as to achieve good ensemble results [16]. This also applies to AdaBoost. For AdaBoost, it is known that there exists a dilemma between weak learner's accuracy and diversity [6], which means that the more accurate two weak learners become, the less they can disagree with each other. Hence, how to select SVM weak learners for AdaBoost? Select accurate but not diverse weak learners? Or select diverse but not too accurate ones? In the proposed AdaBoostSVM, the obtained SVM weak learners are mostly moderately accurate, which give chances to obtain more un-correlated weak learners. As aforementioned, through adjusting the σ value, a set of SVM weak learners with different learning abilities is obtained. This provides an opportunity of selecting more diverse weak learners from this set to deal with the accuracy/diversity dilemma. Hence, we proposed a *Diverse AdaBoostSVM algorithm* (Table. III), and it is hoped to achieve higher generalization performance than AdaBoostSVM.

B. Diverse AdaBoostSVM

Although how to measure and use diversity for ensemble methods is still an open problem [10], recently some promising results have been reported. By focusing on increasing diversity, these methods [9][17] achieved higher generalization accuracy. In the proposed Diverse AdaBoostSVM algorithm, we will use the definition of diversity in [9], which measures the disagreement between one weak learner and all the existing weak learners. In the Diverse AdaBoostSVM, the diversity is calculated as follows: If $h_t(\mathbf{x}_i)$ is the prediction label of t-th weak learner on sample $\mathbf{x}_i$, and $f(\mathbf{x}_i)$ is the combined prediction label of all the existing weak learners, the diversity of the t-th weak learner on sample $\mathbf{x}_i$ is defined as:

$$d_t(\mathbf{x}_i) = \begin{cases} 0: & \text{if } h_t(\mathbf{x}_i) = f(\mathbf{x}_i) \\ 1: & \text{if } h_t(\mathbf{x}_i) \neq f(\mathbf{x}_i) \end{cases} \tag{7}$$

and the diversity of AdaBoostSVM with T weak learners on N samples is defined as:

$$D = \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} d_t(\mathbf{x}_i) \tag{8}$$

In each cycle of Diverse AdaBoostSVM, this diversity value is calculated first. If this value is larger than the predefined threshold, DIV, this new RBFSVM weak learner will be selected. Otherwise, this weak learner will be thrown away. Through this mechanism, a set of moderately accurate and yet diverse SVM weak learners can be generated. This is different from the AdaBoostSVM which simply takes all the available SVM weak learners. As seen from the following experimental results, the Diverse AdaBoostSVM algorithm gives the best performance. We think that the improvement is due to its explicitly dealing with the accuracy/diversity dilemma.

TABLE III

```
Algorithm: Diverse AdaBoostSVM
1. Input: a set of training samples with labels $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$; the initial $\sigma$, $\sigma_{ini}$; the minimal $\sigma$, $\sigma_{min}$; the step of $\sigma$, $\sigma_{step}$; the threshold on diversity $DIV$.
2. Initialize: the weight of samples: $w_i^1 = 1/N$, for all $i = 1, ..., N$.
3. Do While ($\sigma > \sigma_{min}$)
    (1)Use RBFSVM algorithm to train the weak learner $h_t$ on the weighted training sample set.
    (2)Calculate training error of $h_t$ : $\epsilon_t = \sum_{i=1}^{N} w_i^t, y_i \neq h_t(\mathbf{x}_i)$.
    (3)Calculate diversity of $h_t$: $D_t = \sum_{i=1}^{N} d_t(\mathbf{x}_i)$.
    (4)If $\epsilon_t > 0.5$ or $D_t < DIV$, decrease $\sigma$ by $\sigma_{step}$ and goto (1).
    (5)Set weight of weak learner $h_t$ : $\alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$.
    (6)Update training samples' weights: $w_i^{t+1} = \frac{w_i^t \exp\{-\alpha_t y_i h_t(\mathbf{x}_i)\}}{C_t}$
    where $C_t$ is a normalization constant, and $\sum_{i=1}^{N} w_i^{t+1} = 1$.
4. Output: $f(\mathbf{x}) = sign(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}))$.
```

V. EXPERIMENTAL RESULT

Since AdaBoost with neural networks weak learners performs better than those with decision trees weak learners [7], we compare proposed AdaBoostSVM and Diverse AdaBoostSVM algorithms with AdaBoost with neural networks weak learners. A large scale of experiments are generated on 13 benchmark data sets and an unbalanced data sets.

A. Benchmark Data Sets

1) Data set information and parameter setting: The 13 benchmark data sets come from UCI Repository, DELVE, and STATLOG. The dimensions of these data sets range from 2 to 60, the numbers of training samples range from 140 to 1300, the numbers of test samples range from 75 to 7000. Detailed information of these data sets can be found at [18] and all of these data sets can be downloaded there. For each data set, 100 partitions are generated into training and test sets. On each partition, for each algorithm, a classifier is trained and then test error is calculated. The final performance of each algorithm on a data set is its average performance over these 100 partitions of this data set.

It is known that σ mainly affects the performance of RBFSVM classifier compared with C. Hence, C is set as a value within 10 to 100 in proposed algorithms. The σ_{min} is set as the average minimal distance between any two training samples and the σ_{ini} is set as about 10 to 15 times of the σ_{min} value or the scatter radius of the training samples in the input space. Although the value of σ_{step} can affect the number of AdaBoostSVM's learning cycles, it has less effect on the final generalization performance, as shown later. Therefore, σ_{step} is set as a value within 1 to 3. The 'DIV' value in the Diverse AdaBoostSVM algorithm is set as 0.7 to 1 times of the maximal diversity value obtained in previous cycles. "0.7" is used as the tolerance of the possible variation of diversity.

2) General performance: In Table IV, the average generalization performance (with standard deviation) for the four algorithms is shown. The test errors of AdaBoost with Neural Networks weak learners (AdaBoost$_{NN}$) and SVM are directly obtained from [8]. From these results, it can be found that proposed AdaBoostSVM algorithm (AdaBoost$_{SVM}$) performs better than AdaBoost$_{NN}$ algorithm, and is comparable to SVM algorithm. Proposed Diverse AdaBoostSVM (Diverse AdaBoost$_{NN}$) performs a little better than SVM algorithm.

TABLE IV

Data set	AB$_{NN}$	AB$_{SVM}$	Diverse AB$_{SVM}$	SVM
Banana	12.3±0.7	12.1±1.7	**11.3±1.4**	11.5±0.7
B. Cancer	30.4±4.7	25.5±5.0	**24.8±4.4**	26.0±4.7
Diabetes	26.5±2.3	24.8±2.3	24.3±2.1	**23.5±1.7**
German	27.5±2.5	23.4±2.1	**22.3±2.1**	23.6±2.1
Heart	20.3±3.4	15.5±3.4	**14.9±3.0**	16.0±3.3
Image	2.7±0.7	2.7±0.7	**2.4±0.5**	3.0±0.6
Ringnorm	1.9±0.3	2.1±1.1	2.0±0.7	**1.7±0.1**
F. Solar	35.7±1.8	33.8±1.5	33.7±1.4	**32.4±1.8**
Splice	**10.1±0.5**	11.1±1.2	11.0±1.0	10.9±0.7
Thyroid	4.4±2.2	4.4±2.1	**3.7±2.1**	4.8±2.2
Titanic	22.6±1.2	22.1±1.9	**21.8±1.5**	22.4±1.0
Twonorm	3.0±0.3	2.6±0.6	**2.5±0.5**	3.0±0.2
Waveform	10.8±0.6	10.3±1.7	10.2±1.2	**9.9±0.4**
Average	16.0±1.6	14.6±1.9	**14.2±1.7**	14.5±1.5

Comparison among four algorithms: AdaBoost with neural networks weak learners (AB$_{NN}$), proposed AdaBoostSVM (AB$_{SVM}$), proposed Diverse AdaBoostSVM (Diverse AB$_{SVM}$) and SVM

3) Influence of C and σ_{ini}: In order to investigate the influence of parameter C on the proposed algorithms, we vary the value of C from 1 to 100, and perform experiments on

100 partitions of the "Titanic" data set of UCI benchmark to obtain their average performances. Figure. 1 shows the comparison result. It can be found that in a considerable scale (in this case, it is from 1 to 100), the variation of C has little effect (less than 1%) on the final generalization performance of proposed algorithm. This is also consistent with the analysis of RBFSVM that C value has less effect on the performance of RBFSVM. Note that the σ value decreases from σ_{ini} to σ_{min} as the number of SVM weak learners increases (See the label of horizontal axis). The small platform at the left top corner of this figure means that the test error does not decrease until the σ decreases to a certain value. Then, the test error decreases quickly to the lowest value and keeps this value with a little variation. This shows that the σ_{ini} value does not have much impact on the final performance of AdaBoostSVM. Furthermore, the learning cycle will be truncated when σ reaches the given σ_{min} value. This makes it possible to truncate the learning cycle to handle the overfitting problem by setting a suitable σ_{min}. This will be investigated in our future work. This property is more valuable because when to stop AdaBoost's learning cycle is still an open problem for AdaBoost.

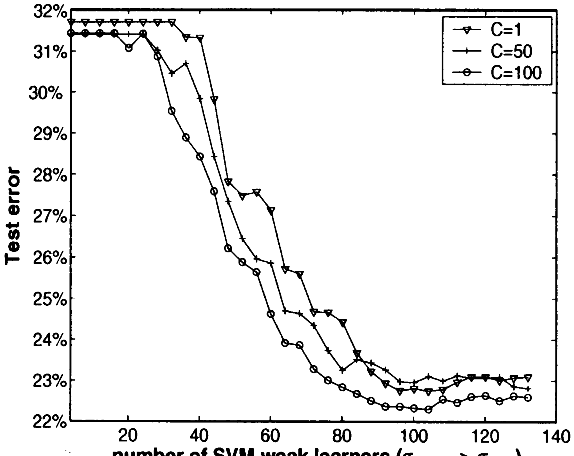


Fig. 1. Compare the performance of AdaBoostSVM with different C values

4) *Influence of σ_{step}:* In order to see the influence of σ_{step} on the proposed AdaBoostSVM algorithm, we did a set of experiments with different σ_{step} values for AdaBoostSVM on the "Titanic" data set of UCI benchmark. Figure. 2 gives the result. From this figure, we can find that although the number of learning cycles in AdaBoost changes with the value of σ_{step}, the final generalization accuracy is relatively stable. Similar conclusions can also be drawn from other benchmark data sets that the value of σ_{step} has less effect on the final generalization performance of AdaBoostSVM.

From the above discussion of parameters C, σ_{ini}, σ_{step}, it can be concluded that, compared with AdaBoost with neural networks weak learners, proposed AdaBoostSVM algorithm is easier to do parameter tuning.

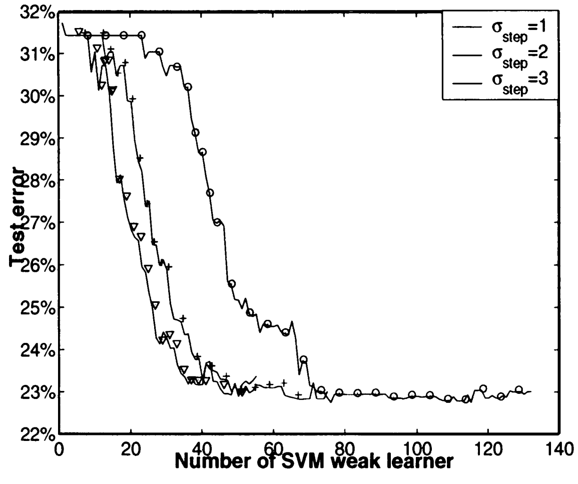
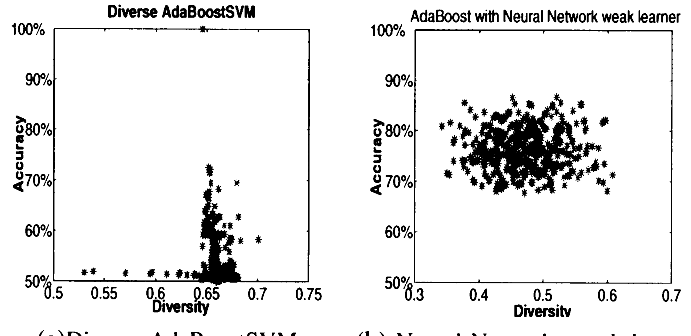


Fig. 2. Compare the performance of AdaBoostSVM with different σ_{step}

5) *Accuracy-Diversity diagram:* In order to see the effect of Diverse AdaBoostSVM, we used an Accuracy-Diversity diagram to visualize the distribution of accuracy and the diversity over the SVM weak learners. The Accuracy-Diversity diagram is a scatterplot where each point corresponds to a weak learner. In this diagram, a point's x coordinate value is the diversity value after the corresponding weak learner is trained while y coordinate value of this point is the accuracy rate of the corresponding weak learner. Similar diagram was also used in [6]. Due to the lack of space, we report only the Accuracy-Diversity diagrams of Diverse AdaBoostSVM and AdaBoost with neural networks weak learner algorithms on the 'Splice' data set of the UCI benchmark. From Figure. 3, we can observe that proposed Diverse AdaBoostSVM algorithm can obtain more high-diversity and moderately accurate weak learners, which may produce the uncorrelated error among different weak learners more efficiently.



(a)Diverse AdaBoostSVM (b) Neural Networks weak learner

Fig. 3. Compare the accuracy and the diversity's distributions between Diverse AdaBoostSVM and AdaBoost with Neural Networks weak learners

B. Unbalanced Data Sets

We also find that the proposed AdaBoostSVM algorithm performs much better than a single SVM when facing unbalanced classification problems. In the case of binary classification, unbalanced problem means that the number of positive samples is much larger than that of negative samples, or vice versa. It is known that standard SVM algorithm cannot handle such kind of problems well. Proposed AdaBoostSVM algorithm includes many moderately accurate SVM weak learners, some of which can focus on the misclassified samples of the non-dominant class. By combining these moderately accurate SVM classifiers, AdaBoostSVM performs better than standard SVM on unbalanced problems. In the following experiments, we compare the performance of proposed AdaBoostSVM and standard SVM on unbalanced data sets. We used the 'Splice' data set of the UCI benchmark. Splice data set has 483 positive training samples, 517 negative training samples and 2175 test samples. In the following experiments, the number of negative samples is fixed as 500 and the number of positive samples is reduced from 150 to 30. From Figure. 4, it can be found that along with the decreasing ratio of the number positive samples to that of negtive samples, the improvement of AdaBoostSVM over SVM becomes more and more. When the ratio reaches 30:500, SVM almost cannot work and performs like random guess, while proposed AdaBoostSVM can still work with a relatively good performance. The good performance of ensemble methods to handle unbalanced problems was also observed in [19] [20].

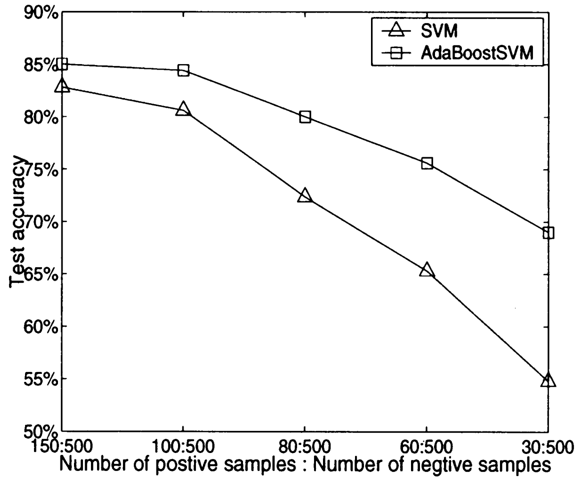


Fig. 4. Compare SVM and AdaBoostSVM on unbalanced problem

VI. CONCLUSION AND DISCUSSION

AdaBoost with SVM weak learners is proposed in this paper, which is achieved by adaptively adjusting the kernel parameter to get a set of effective weak learners. Experiments on benchmark data sets demonstrated that proposed AdaBoostSVM performs better than AdaBoost with neural networks weak learners. In addition, it needs not accurate model selection and thus saves computational cost. Based on AdaBoostSVM, an improved version is further developed to deal with the accuracy/diversity dilemma, and promising result is obtained. Besides these, it is found that proposed AdaBoostSVM algorithm has much better performance than SVM on unbalanced problems. The reported in this paper is a general framework of AdaBoost with SVM weak learners, which can be tailored for different purposes through changing the parameter scales or selection criterion. In the future work, more theoretical analysis and comparison with AdaBoost with all kinds of weak learners will be conducted.

REFERENCES

[1] Robert E. schapire, "The boosting approach to machine learning: An overview," *In MSRI Workshop on Nonlinear Estimation and Classification*, 2002.

[2] Leo Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[3] Eric Bauer and Ron Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning*, vol. 36, no. 1, pp. 105–139, Jul 1999.

[4] 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*, vol. 55(1), pp. 119–139, August 1997.

[5] R. E. Schapire, Y. Singer, P. Bartlett, and W. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *The Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.

[6] Thomas G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine Learning*, vol. 40, no. 2, pp. 139–157, Aug 2000.

[7] Holger Schwenk and Yoshua Bengio, "Boosting neural networks," *Nueral Computation*, vol. 12, pp. 1869–1887, 2000.

[8] Gunnar Ratsch, "Soft margins for adaboost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, Mar 2001.

[9] Prem Melville and Raymond J. Mooney, "Creating diversity in ensembles using artificial data," *Information Fusion*, vol. 6, no. 1, pp. 99–111, Mar 2005.

[10] Ludmila I. Kuncheva and Christopher J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, May 2003.

[11] Wenxin Jiang, "Process consistency for adaboost," *Annals of Statistics*, vol. 32, no. 1, pp. 13–29, 2004.

[12] Vladimir Vapnik, *Statistical Learning Theory*, John Wiley and Sons Inc., New York, 1998.

[13] Bernhard Scholkopf, Kah-Kay Sung, Chris Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik, "Comparing support vector machines with gaussian kernels to radial basis function classifiers," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2758–2765, 1997.

[14] Giorgio Valentini and Thomas G. Dietterich, "Bias-variance analysis of support vector machines for the development of svm-based ensemble methods," *Journal of Machine Learning Research*, vol. 5, pp. 725–775, Jul 2004.

[15] Robert E. Schapire and Yoram Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, Dec 1999.

[16] H.W.Shin and S.Y.Sohn, "Selected tree classifier combination based on both accuracy and error diversity," *Pattern Recognition*, vol. 38, pp. 191–197, 2005.

[17] Sanjoy Dasgupta and Philip M. Long, "Boosting with diverse base classifiers," in *Proceeding of the 16th Annual Conference on Learning Theory*, Aug 2003, pp. 273–287.

[18] `http://mlg.anu.edu.au/~raetsch/data`.

[19] Rong Yan, Yan Liu, Rong Jin, and Alex Hauptmann, "On predicting rare class with svm ensemble in scene classification," in *Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal 2003*, Apr 2003, pp. III - 21–4.

[20] Hyun-Chul Kim, Shaoning Pang, Hong-Mo Je, Daijin Kim, and Sung Yang Bang, "Constructing support vector machine ensemble," *Pattern Recognition*, vol. 36, no. 12, pp. 2757–2767, Dec 2003.