Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

MCs Detection Approach Using Bagging and Boosting Based Twin Support Vector Machine

Xinsheng Zhang
School of management, Xi'an Univ. of Arch. and Tech.
School of Electronic Engineering, Xidian University
Xi'an, China
xinsheng.zh@hotmail.com

Xinbo Gao
School of Electronic Engineering
Xidian University
Xi'an, China
xbgao@lab202.xidian.edu.cn

Minghu Wang
School of Automatic Control
Northwestern Polytechnical University
Xi'an, China
wangminghuemail@163.com

***Abstract*—In this paper we discuss a new approach for the detection of clustered microcalcifications (MCs) in mammograms. MCs are an important early sign of breast cancer in women. Their accurate detection is a key issue in computer aided detection scheme. To improve the performance of detection, we propose a Bagging and Boosting based twin support vector machine (BB-TWSVM) to detect MCs. The algorithm is composed of three modules: the image pre-processing, the feature extraction component and the BB-TWSVM module. The ground truth of MCs in mammograms is assumed to be known as a priori. First each MCs is preprocessed by using a simple artifact removal filter and a well designed high-pass filter. Then the combined image feature extractors are employed to extract 164 image features. In the combined image features space, the MCs detection procedure is formulated as a supervised learning and classification problem, and the trained BB-TWSVM is used as a classifier to make decision for the presence of MCs or not. The experimental results of this study indicate the potential of the approach for computer-aided detection of breast cancer.**

***Keywords*—enseble learning, Boosting, feature extraction, detection, clustered microcalcifications, Bagging, twin support vector machine**

I. INTRODUCTION

Brest cancer is the most frequently diagnosed cancer in women[1, 2]. Digital mammography is, at present, one of the most suitable methods for early detection of breast cancer[3]. One of the important early signs of breast cancer is the appearance of clustered microcalcifications (MCs), which appear in 30% -50% of mammographically diagnosed cases with tiny bright spots of different morphology. Because of its importance in early breast cancer diagnosis, accurate detection of MCs has become a key problem.

Recently, a lot of different approaches based on machine learning have been applied to the detection of MCs [4]. Concerning image segmentation and specification of regions of interest(ROI), several methods have been reviewed in [5]. Formulated MCs detection as a classification problem, various machine learning methodologies have been proposed for the characterization of MCs, such as, fuzzy rule-based systems [6], support vector machines [7, 8], neural networks [7, 9, 10], etc. These methods learn hypotheses from a large amount of diagnosed samples, i.e., the data collected from a number of necessary medical examinations along with the corresponding diagnoses made by medical experts. To make these approaches work well, a large amount of positive (with suspicious region) and negative (with normal tissue) samples with diagnosis are needed for training the learning model. Usually, these negative samples can be easily collected and the number is also very large, which is always enough for training any learning model. However, the positive samples are often very small. It is because that making a diagnosis for such a large amount of cases one by one places a very heavy burden on medical experts. To solve the problem, one possible solution is to learn a hypothesis from the limited number of the positive and negative samples. In machine learning, this problem is called learning with unbalanced data. Recently Jayadeva *et al.* [11] proposed a nonparallel plane classifiers for binary data classification, termed as twin support vector machine (TWSVM). It aims at generating two nonparallel planes, which use two different group of samples, such that each plane is closer to one of the two classes and is as far as possible from the other. TWSVM solves a pair of quadratic programming problems (QPPs). The two QPPs have a smaller size than the one larger QPP, which makes TWSVM work faster than the standard support vector machine(SVM). And also TWSVM has the potential ability to deal with the unbalanced data sets by solving the two different smaller size problems. But the same as SVM, TWSVM is unstable, because it is sensitive to the training samples. To eliminate the underlying weakness of TWSVM, we design a novel mechanism for MCs detection -- Bagging and Boosting based twin support vector machine (BB-TWSVM), which

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

incorporates Bagging and Boosting with TWSVM. The key idea comes from the ensemble learning or multiple classifier system [12-14].

Ensemble methods, well known in the machine learning community, are typically composed of multiple methods comprising different classification strategies or different classifiers with a unified objective function. The final predictions or decisions are chosen from the ensemble of methods by a learning rule, which may be as simple as finding the maximum score from all the base learners, or as complex as optimizing a weighted scoring scheme from among the base methods. The construction of this learning rule is a key problem to the performance of an ensemble learning method, as the performance of an ensemble method with an ineffective learning rule will be the average of the performance of its component algorithms.

To improve the performance of the available MCs detection algorithms, this paper employs the ensemble learning method with Bagging and Boosting based twin support vector machine (BB-TWSVM) as a final decision model to distinguish the MCs from the other ROIs (region of interests). BB-TWSVM is first trained by the real mammograms from DDSM. Then it is used to detect other ROIs. Compared with single learning algorithm and other existing ensemble methods, the proposed approach yields superior performance when evaluated using receiver operating characteristic (ROC) curves. It achieved average sensitivity as high as 94.09% with about 6.31% average false-positive rate in each testing phrase.

The rest of the paper is organized as follows. The proposed BB-TWSVM algorithm is formulated in Section II. Section III reports the feature extraction methods. The proposed MCs detection algorithm based on BB-TWSVM is formulated in Section IV. and the experimental results are given in Section V. Finally, conclusions are drawn in Section VI.

II. Boosting And Bagging Based TWSVM

A. Ensemble Learning

An ensemble consists of a set of individually trained classifiers, such as neural networks, support vector machines, decision trees etc., whose predictions are combined when classifying the inputs [11]. Ensemble learning combines multiple learned models or classifiers under the assumption that "two (or more) heads are better than one". Previous research has shown that an ensemble is often more accurate than any of the single classifiers in the ensemble. An ensemble combines different base learning algorithms or the same learning algorithms trained in the same or different ways. As we know, an ensemble is a group of classifiers that jointly solve a classification problem. Both theoretical and empirical findings support that an ensemble will often outperform a single classifier. In addition, the ensemble learning framework provides tools to tackle complicated or large problems that were once infeasible for traditional algorithms. In view of the advantages of an ensemble against a single classifier, many ensemble generation algorithms have been presented during the past decade to systematically construct collective classifiers that as a whole can achieve better performance than a single classifier. Bagging and Boosting are two most famous algorithms to construct an ensemble classifier.

B. Twin Support Vector Machine

Twin support vector machine (TWSVM) is a recently developed and very effective binary classification algorithm, which obtains nonparallel planes around which the data points of the corresponding class get clustered. Each of the two quadratic programming problems in a TWSVM pair has the formulation of a typical SVM, except that not all patterns appear with constrains of either problem at the same time.

The TWSVM classifier is obtained by solving the following pair of quadratic programming problems:

$$\text{(TWSVM1)} \quad \min_{\mathbf{w}^{(1)},\, b^{(1)},\, \mathbf{q}} \quad \frac{1}{2}(\mathbf{A}\mathbf{w}^{(1)}+\mathbf{e}_1 b^{(1)})^T(\mathbf{A}\mathbf{w}^{(1)}+\mathbf{e}_1 b^{(1)})+c_1\mathbf{e}_2^T\mathbf{q}, \text{ and} \quad (1)$$
$$\text{s.t.} \quad -(\mathbf{B}\mathbf{w}^{(1)}+\mathbf{e}_2 b^{(1)})+\mathbf{q}\geq \mathbf{e}_2, \quad \mathbf{q}\geq 0$$

$$\text{(TWSVM2)} \quad \min_{\mathbf{w}^{(2)},\, b^{(2)},\, \mathbf{q}} \quad \frac{1}{2}(\mathbf{B}\mathbf{w}^{(2)}+\mathbf{e}_2 b^{(2)})^T(\mathbf{B}\mathbf{w}^{(2)}+\mathbf{e}_2 b^{(2)})+c_2\mathbf{e}_1^T\mathbf{q} \quad (2)$$
$$\text{s.t.} \quad (\mathbf{A}\mathbf{w}^{(2)}+\mathbf{e}_1 b^{(2)})+\mathbf{q}\geq \mathbf{e}_1, \quad \mathbf{q}\geq 0,$$

where $c_1, c_2 > 0$ are parameters and $\mathbf{e}_1$ and $\mathbf{e}_2$ are vectors of ones of appropriate dimensions.

This approach aims to find two optimal hyperplanes, one for each class, and classifiers points according to which hyperplane a given point is closest to. The first term in the objective function of (1) or (2) is the sum of squared distances from the hyperplane to points of one class. Therefore, minimizing it tends to keep the hyperplane close to points of one class (i.e., class 1). The constraints require the hyperplane to be at a distance of at least 1 from points of the other class (i.e., class -1); a set of error variables is used to measure the error wherever the hyperplane is closer than this minimum distance of 1. The second term of the objective function minimizes the sum of error variables, thus attempting to minimize misclassification due to points belonging to class -1. As an example in Fig. 1, one can find the difference between traditional SVM and TWSVM. Similarly as SVM, TWSVM can be easily extended to nonlinear kernel version.

C. Bagging Twin Support Vector Machine

Bagging is a bootstrap ensemble method that creates individuals for its ensemble by training each classifier on a random redistribution of the training set. It incorporates the benefits of bootstrapping and aggregation. The final multiple classifiers can be generated by training on multiple sets of samples that are produced by bootstrapping, i.e., random sampling with replacement on the training samples. Aggregation of the generated classifiers can then be implemented by majority voting or weighted averaging.

Experimental and theoretical results have shown that Bagging can improve a good but unstable classifier significantly. This is exactly the problem of TWSVM. However, directly using Bagging in TWSVM is not appropriate since we have only a very small number of samples. To overcome this problem, we develop a novel Bagging strategy. The bootstrapping is executed only on the negative samples since there are far more negative samples than the positive samples. This way each generated classifier will be trained on a balanced number of positive and negative samples. The Bagging TWSVM algorithm is described in Table I. In the algorithm, the aggregation is implemented by weighted averaging.

TABLE I. ALGORITHM OF BAGGING TWIN SUPPORT VECTOR MACHINE

Input: positive training set S^+, negative training set S^-, weak classifier I (TWSVM), integer T (number of generated classifiers), and the test sample $X_i (i=1,...,m)$.
for $t = 1$ to T {
S_t^- = bootstrap samples from S^-, where $\lvert S_t^- \rvert = \lvert S^+ \rvert$
$C_t = I(S_t^-, S^+)$
$w_t = C_t(S_t^-, S^+)$
}
$C^*(X) = aggregation\{C_t(X, S_t^-, S^+), 1 \le t \le T\}$
Output: final classifier C^*.

D. *Boosting Twin Support Vector Machine*

Boosting is a machine learning meta-algorithm for performing supervised learning. Boosting is based on the question posed by Kearns[15]: can a set of weak learners create a single strong learner? A weak learner is defined to be a classifier which is only slightly correlated with the true classification. In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification.

While Boosting is not algorithmically constrained, most Boosting algorithms consist of iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier. When they are added, they are typically weighted in some way that is usually related to the weak learner's accuracy. After a weak learner is added, the data is reweighted: examples that are misclassified gain weight and examples that are classified correctly lose weight (some Boosting algorithms actually decrease the weight of repeatedly misclassified examples, e.g., boost by majority and BrownBoost). Thus, future weak learners focus more on the examples that previous weak learners misclassified.

Experimental and theoretical results have shown that Boosting can improve a good but unstable classifier significantly. This is exactly the problem of TWSVM. However, directly using Boosting in TWSVM is not appropriate since we have only a very small number of samples. To overcome this problem, we develop a novel Bagging strategy. The bootstrapping is executed only on the negative samples since there are far more negative samples than the positive samples. This way each generated classifier will be trained on a balanced number of positive and negative samples. The Boosting TWSVM algorithm is described in Table II.

E. *Bagging and Boosting based TWSVM*

In order to combine the advantages of the above two algorithms, we formulate them into one approach-- Bagging and Boosting based TWSVM.To In short, the Bagging and Boosting based TWSVM algorithm performs a bootstrap aggregation on the boosted weak learner (TWSVM). The parameter selection is fundamental: T is the chosen number of the bootstrap replicates, M is the number of iterations of the Boosting algorithm. The sub-sample consists of a fraction ρ of the training set. Notice that we do not allow Boosting to run for hundreds of iterations. This is a fundamental feature of our algorithm and it arises from the observation that Boosting is a self-smoothing algorithm. We shall show that running Boosting for many iterations, well past the sample of achieving zero training set error rate, allows the algorithm to smooth itself. Since the smoothing in this algorithm is accomplished directly by Bagging the large number of iterations are not needed and can be limited to a fixed number.

TABLE II. ALGORITHM OF BOOSTING TWIN SUPPORT VECTOR MACHINE

Input: positive training set S^+, negative training set S^-, weak classifier I (TWSVM), integer T (number of generated classifiers), the test sample $X_i (i=1,...,N)$, distribution D over the N examples
Initialize the weight vector: $w_i^1 = D(i)$ for i=1, …, N.
Do for t = 1, 2, …, T
1. Set $\mathbf{p}^t = \dfrac{\mathbf{w}^t}{\sum_{i=1}^{N} w_i^t}$
2. Call Weak classifier, providing it with the distribution $\mathbf{p}^t$; get back a hypothesis $h_t : \mathrm{X} \rightarrow [0,1]$.
3. Calculate the error of $h_t : \varepsilon_t = \sum_{i=1}^{N} p_i^t \lvert h_t(x_i) - y_i \rvert$.
4. Set $\beta_t = \varepsilon_t / (1-\varepsilon_t)$.
5. Set the new weights vector to be $w_i^{t+1} = w_i^t \beta_t^{1-\lvert h_t(x_i) - y_i \rvert}$
Output the hypothesis $C^*(X) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} (\log 1/\beta_t) h_t(x) \ge \frac{1}{2} \sum_{t=1}^{T} \log 1/\beta_t \\ 0 & \text{otherwise}. \end{cases}$

TABLE III. ALGORITHM OF BOOSTING AMD BAGGING BASED TWIN SUPPORT VECTOR MACHINE

Input: positive training set S^+, negative training set S^- ($S = S^+ \cup S^-, \lvert S \rvert = N$), weak classifier I (TWSVM), (Bagging) integer T (number of generated classifiers) and $\rho > 0$, (Boosting) parameter M.
Do for t = 1, 2, …, T
1. Generate a replicate training set $S_t^{'}$ from S^+ and S^- of size $S_k^{'} = \rho N$ by sub-sampling with replacement.
2. Generate a boosted classifier, $C_t^*(X)$, based on $S_t^{'}$ using M *iterations of the base learner*.
Output the aggregate classifier C^*:
$C^*(X) = \arg\max_{y \in Y} \sum_{t: C_t^*(X)} 1$

III. FEATURE EXTRACTION

The research methodology uses a novel Bagging and Boosting based twin support vector machine to detect MCs with a set of combined image features from a number of existing features. As we know, feature extraction is a very important part for the classification problem. To get the best feature or combination of features and get the high classification rate for MCs detection is one of main aims of the proposed research. In our task, a set of 164 features, shown in

Table IV, is calculated for each suspicious area (ROI) from the textural, spatial and transform domains in our research.

Before training the classifier, we use the feature extractor discussed in Table IV to extract MCs features in the feature domain. The 164 features of each block are extracted as follows.

- Intensity histogram based texture feature

A frequently used approach for texture analysis is based on statistical properties of the intensity histogram. One class of such measures is based on statistical moments. The expression for the *n*th moment about the mean is given by

$$\mu_n = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i) \tag{3}$$

where z_i is a random variable indicating intensity, $p(z)$ is the histogram of the intensity levels in a region, L is the number of possible intensity levels, and

$$m = \sum_{i=0}^{L-1} z_i p(z_i) \tag{4}$$

is the mean(average)intensity. Table V lists the descriptors used in our experiments based on statistical moments and also on uniformity and entropy.

TABLE IV. COMBINED DIFFERENT KIND OF IMAGE FEATURES USED IN OUR EXPERIMENTS

Feature groups	Type	#No.
Histogram based texture features	Mean	1
	Standard deviation	2
	Smoothness	3
	Third moment	4
	Uniformity	5
	Entropy	6
Multi-scale histogram features	Histograms with different number of bins 3,5,7,9	7~30(total 24)
Zernike Moments Features[16]	Zernike Moments	31~66 (total 36, D=10)
Combined first 4 moments features	Combined moments feature	67~114 (total 48)
Tamura texture signatures	Tamura features	115~120 (total 6)
Chebyshev transform histogram feature [17]	Chebyshev histogram	121~152 (total 32)
Radon Transform Features	Radon features	153~164 (total 12)

TABLE V. DESCRIPTORS OF THE TEXTURE BASED ON THE INTENSITY HISTOGRAM OF A REGION

Moment	Formula	#No.
Mean	$m = \sum_{i=0}^{L-1} z_i p(z_i)$	1
Standard deviation	$\sigma = \sqrt{\mu_2(z)} = \sqrt{\sigma^2}$	2
Smoothness	$R = 1 - 1/(1+\sigma^2)$	3
Third moment	$\mu_3 = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i)$	4
Uniformity	$U = \sum_{i=0}^{L-1} p^2(z_i)$	5
Entropy	$e = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i)$	6

- Multiscale intensity histograms

We compute signatures based on "multi-scale histograms" idea. Idea of multi-scale histogram comes from the belief of a unique representation of an image through infinite series of histograms with sequentially increasing number of bins. Here we used 4 histograms with number of bins being 3,5,7,9 to calculate the image feature, so we get a 1*24 row vectors.

- Zernike moments

We use derived moments based on alternative complex polynomial functions, know as Zernike polynomials[16]. They form a complete orthogonal set over the interior of the unit circle $x^2 + y^2 = 1$ and are defined as

$$Z_{pq} = (p+1)/\pi \int_{x^2+y^2 \le 1} f(x,y) V(\rho,\theta) dxdy, \tag{5}$$

$$V_{pq}(x,y) = V_{pq}(\rho,\theta) = R_{pq}(\rho)\exp(jq\theta), \tag{6}$$

$$R_{pq}(\rho) = \sum_{s=0}^{(p-|q|)/2} \frac{(-1)^s[(p-s)!]\rho^{p-2s}}{s!\left(\frac{p+|q|}{2}-s\right)!\left(\frac{p-|q|}{2}-s\right)!}, \tag{7}$$

where p is a nonnegative integer, q is an integer subject to the constraint $p-|q| = \text{even}$ and $|q| \le p$, $\rho = \sqrt{x^2+y^2}$ is the radius from (x,y) to the image centroid, $\theta = \tan^{-1}(y/x)$ is the angle between ρ and x-axis. The Zernike moment Z_{pq} is order p with repetition q . For a digital image, the respective Zernike moments are computed as

$$Z_{pq} = (p+1)/\pi \sum_i f(x_i,y_i) V(\rho,\theta) dxdy, x^2+y^2 \le 1, \tag{8}$$

where i runs over all the image pixels. Zernike moments are used as the feature extractor where by the order is varied to achieve the optimal classification performance.

- Combined first four moment features

Signatures on the basis of first four moments (also known as mean, std, skewness, kurtosis) for data generated by vertical, horizontal, diagonal and alternative diagonal 'combs'. Each column of the comb results in 4 scalars [mean, std, skewness, kurtosis], we have as many of those [...] as 20. So, 20 go to a 3-bin histogram, producing 1x48 vectors.

- Tamura texture signatures

Tamura et al. [17] took the approach of devising texture features that correspond to human visual perception. Six textural features: coarseness, contrast, directionality, line-likeness, regularity and roughness, are defined for the image feature for object recognition.

- Transform domain features

We computes signatures (32 bins histogram) from coefficients of 2D Chebyshev transform (10th order is used in our experiments). Also we used signatures based on the Radon transform as a kind of image features. Radon transform is the projection of the image intensity along a radial line (at a specified orientation angle), total 4 orientations are taken. Transformation n/2 vectors (for each rotation), go through 3-bin histogram and convolve into 1x12 vectors.

IV. BAGGING AND BOSSTING BASEDTWSVM FOR MCS DETECTION

For a given digital mammography image, we consider the MCs detection process as the following steps:

- Step 1. Preprocess the mammography image by removing the artifacts, suppressing the inhomogeneity of the background and enhancing the microcalcifications.
- Step 2. At each pixel location in the image, extract a

$A_{m\times m}$ small window to describe its surrounding image feature.

- Step 3. Apply feature extraction methods to get the combined feature vector $\boldsymbol{x}$.
- Step 4. Use the trained BB-TWSVM classifier to make decision whether $\boldsymbol{x}$ belongs to MCs class (+1) or not (−1).

A. Mammogram Preprocessing

First, a simple film-artifact removal filter is applied to remove the film-artifacts from mammography image. After the artifacts were removed, our task is to suppress the mammogram background. A high-pass filter is designed to preprocess each mammogram before extracting samples. With the Gaussian filter, $f(x,y)=\exp(-(x^2+y^2)/(2\sigma^2))$, we use a $m\times m$ window size Gaussian filter where $m=4\sigma^2+1$, experimentally in the study $\sigma=2$. The output of high-pass filter is denoted by $I_2(x,y)=I_1(x,y)-f(x,y)*I_1(x,y)$, where $*$ is linear convolution.

B. Input Patterns for MCs detection

After the mammogram preprocessing stage, we extract combined image features from $A_{m\times m}$ as a feature vector $\boldsymbol{x}$. $A_{m\times m}$ is a small window of $m\times m$ pixels centered at a location that we concerned in a mammogram image. The choice of m should be large enough to include the MCs (in our experiment, we take m=115). The task of the TWSVM classifier is to decide whether the input window $A_{m\times m}$ at each location is a MCs pattern ($y=+1$) or not ($y=-1$).

C. Training Data Set Preparation

The procedure for extracting training data from the training mammograms is given as follows. For each MCs location in a training mammogram set, a window of $m\times m$ image pixels centered at its center of mass is extracted; the area is denoted by $A^i_{m\times m}$, with respect to $\boldsymbol{x}_i$ after subspace feature extraction, and then $\boldsymbol{x}_i$ is treated as an input pattern for the positive sample ($y_i=+1$). The negative samples are collected ($y_i=-1$) similarly, except that their locations are randomly selected from the non MCs locations in the training mammograms. In the procedure, no window in the training set is allowed to overlap with any other training window.

V. EXPERIMENTAL RESULTS

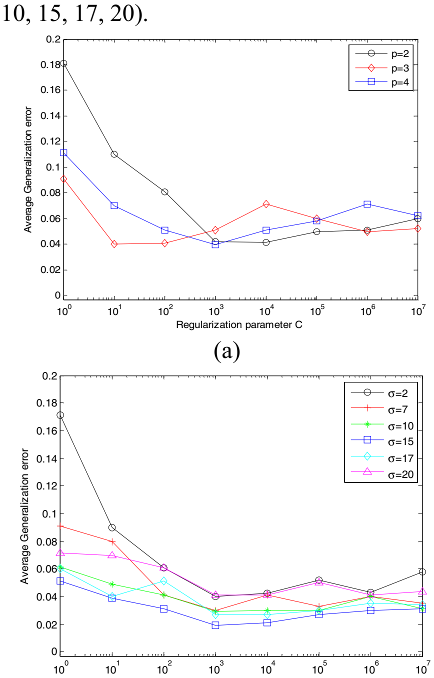
A. Combined Feature Extraction

The data in the training (and validation), test, and validation sets were randomly selected from the training set. Each selected sample was covered by a 115x115 window whose center coincided with the center of mass of the suspected MCs. The blocks included 65 with true MCs and 3567 with normal tissue.

Before training the classifier, we first use the feature extraction algorithm to extract MCs feature in combined feature domain. The 164-dimension feature vector will be calculated for each image block. When we get the image feature vector, feature normalization program should be user for normalizing the features to be real numbers in the range of [0, 1]. The normalization is accomplished by the following step: (a) Change all the features to be positive by adding the magnitude of the largest minus value of this feature times 1.01^2;(b) Divide all the features by the maximum value of the same feature. The normalized features are used as the inputs of the proposed ensemble learning algorithm for training and classification.

B. BB-TWSVM Model Training and Selection

For base model selection, the TWSVM classifier is first trained by using the 10-fold cross-validation procedure with different model and parametric settings. In our training stage, we used generalization error, which was defined as the total number of incorrectly classified examples divided by the total number of samples classified, as a metric to measure the trained classifier. Generalization error was computed using only the samples used during training. For the sake of convenient, the parametric values of c_1 and c_2 in our experiment are set to be equal (i.e. $c_1=c_2$). In Figure 1(a), we summarize the results for trained TWSVM classifier with a polynomial kernel. The estimated generalization error is plotted versus the regularization parameters c_1 and c_2 for kernel order $p=2$, $p=3$ and $p=4$. Similarly, Figure 1(b) summarizes the results with RBF kernel. Here, the generalization error is plotted for different values of the width σ (2, 7, 10, 15, 17, 20).



(a)

(b)

Figure 1. Plot of average generalization error rate versus regularization parameter C ($c_1=c_2$) achieved by training TWSVM using (a) a polynomial kernel with orders 2, 3, and 4, and (b) a Gaussian RBF kernel width σ=2, 7, 10, 15, 17, 20 by 10-*fold* validation method.

TABLE VI. EXPERIMENTAL RESULTS OF BB-TWSVM AND TWSVM CLASSIFIER

Method	Sensitivity	Specificity	Az
TWSVM	90.01%	90.37%	0.9459
BB-TWSVM	94.09%	93.79%	0.9526

From Figure 2, it can be shown that, the BB-TWSVM algorithm has a higher detection accuracy rate compared to TWSVM with the same configuration. By using the same extracted feature, compared with TWSVM, BB-TWSVM has a better detection performance when we train the classifier. In particular, the BB-TWSVM classifier achieved the averaged sensitivity of approximately 94.09% with respect to 6.31% false positive rate and Az=0.9526. With the same training data set and test data set, the TWSVM classifier achieved a sensitivity of 90.01%, 9.63% false positive rate and Az=0.9459.

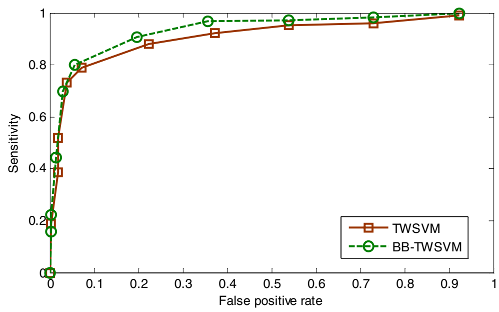


Figure 2. Roc curves of MCS detection using BB-TWSVM and TWSVM classifier, where Az (BB-TWSVM)= 0.9627 and Az (TWSVM)= 0.9459.

VI. CONCLUSION

In this paper, we proposed a Bagging and Boosting based TWSVM (BB-TWSVM) technique for detection MCs in digital mammograms. In this method, combined image features are extracted from each image block of positive and negative samples, and BB-TWSVM is trained through supervised learning with the 164 dimensional feature to test at every location in a mammogram whether an MCs is present or not. The decision function of the trained BB-TWSVM is determined in terms of ensemble classifier model that are modeled from the positive and negative samples during training stage. Compared to TWSVM classifier, BB-TWSVM can solve the unstable problem of TWSVM while maintaining the detection accuracy in a noisy environment. In our experiments, ROC curves indicated that the BB-TWSVM learning approach yielded the better performance compared with TWSVM.

ACKNOWLEDGMENTS

This work presented in our paper is supported by the National Natural Science Foundation of China (No. 60771068), the Natural Science Basic Research Plan in Shaanxi Province of China (No. 2007F$_2$48), the Scientific Research Plan Projects of Shaanxi Education Department (No. 09JK563 and No. 09JK542). The authors are grateful for the anonymous reviewers who made constructive comments.

REFERENCES

[1] American Cancer Society, *Breast Cancer Facts & Figures 2007-2008*, American Cancer Society, Inc, Atlanta, 2007.
[2] A. Jemal, R. Siegel, E. Ward *et al.*, "Cancer Statistics, 2006," *CA: A Cancer Journal for Clinicians,* vol. 56, no. 2, pp. 106-130, 2006.
[3] J. Dengler, S. Behrens, and J. Desaga, "Segmentation of microcalcifications in mammograms," *IEEE Transactions on Medical Imaging,* vol. 12, no. 4, pp. 634-642, 1993.
[4] L. Wei, Y. Yang, R. Nishikawa *et al.*, "A study on several Machine-learning methods for classification of Malignant and benign clustered microcalcifications," *IEEE Transactions on Medical Imaging,* vol. 24, no. 3, pp. 371-380, 2005.
[5] H. Cheng, X. Cai, X. Chen *et al.*, "Computer-aided detection and classification of microcalcifications in mammograms: a survey," *Pattern Recognition,* vol. 36, no. 12, pp. 2967-2991, 2003.
[6] R. Mousa, Q. Munib, and A. Moussa, "Breast cancer diagnosis system based on wavelet analysis and fuzzy-neural," *Expert Systems with Applications,* vol. 28, no. 4, pp. 713-723, 2005.
[7] A. Papadopoulos, D. I. Fotiadis, and A. Likas, "Characterization of clustered microcalcifications in digitized mammograms using neural networks and support vector machines," *Artificial Intelligence in Medicine,* vol. 34, no. 2, pp. 141-150, 2005.
[8] I. El-Naqa, Y. Yongyi, M. N. Wernick *et al.*, "A support vector machine approach for detection of microcalcifications," *IEEE Transactions on Medical Imaging,* vol. 21, no. 12, pp. 1552-1563, 2002.
[9] S. Halkiotis, T. Botsis, and M. Rangoussi, "Automatic detection of clustered microcalcifications in digital mammograms using mathematical morphology and neural networks," *Signal Processing,* vol. 87, no. 7, pp. 1559-1568, 2007.
[10] L. Bocchi, G. Coppini, J. Nori *et al.*, "Detection of single and clustered microcalcifications in mammograms using fractals models and neural networks," *Medical Engineering & Physics,* vol. 26, no. 4, pp. 303-312, 2004.
[11] R. Jayadeva, and S. Chandra, "Twin Support Vector Machines for Pattern Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, no. 5, pp. 905-910, 2007.
[12] Z. Zhi-Hua, and Y. Yang, "Ensembling local learners Through Multimodal perturbation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B,*vol. 35, no. 4, pp. 725-735, 2005.
[13] G. I. Webb, and Z. Zheng, "Multistrategy ensemble learning: reducing error by combining ensemble learning techniques," *IEEE Transactions on Knowledge and Data Engineering,* vol. 16, no. 8, pp. 980-991, 2004.
[14] N. Ueda, "Optimal linear combination of neural networks for improving classification performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 2, pp. 207-215, 2000.
[15] R. Schapire, and Y. Freund, "A decision theoretic generalization of on-line learning and an application to Boosting," *J Comput Syst Sci,* vol. 55, pp. 119-139, 1997.
[16] L. Tsong-Wuu, and C. Yun-Feng, "A comparative study of Zernike moments." in Proceeding of WI2003, 2003 pp. 516-519.
[17] M. S. Tamura H., Yamawaki T., "Texture features corresponsing to visual perception," *IEEE Transactions. on Systems, Man and Cybernetics,,* vol. 1, no. 8, pp. 460-472, 1978.