Learning to Improve Area-Under-FROC for Imbalanced Medical Data Classification Using an Ensemble Method

Hung-Yi Lo, Chun-Min Chang, Tsung-Hsien Chiang, Cho-Yi Hsiao, Anta Huang, Tsung-Ting Kuo, Wei-Chi Lai, Ming-Han Yang, Jung-Jung Yeh, Chun-Chao Yen, Shou-De Lin
Computer Science and Information Engineering Department,
National Taiwan University
Taipei, Taiwan
{d96023, r96944012, b93009, r93100, b93081, d97944007, d96031, r94134, b93032, r96944016, sdlin}@csie.ntu.edu.tw

ABSTRACT
This paper presents our solution for KDD Cup 2008 competition that aims at optimizing the area under ROC for breast cancer detection. We exploited weighted-based classification mechanism to improve the accuracy of patient classification (each patient is represented by a collection of data points). Final predictions for challenge 1 are generated by combining outputs from weighted SVM and AdaBoost; whereas we integrate SVM, AdaBoost, and GA to produce the results for challenge 2. We have also tried location-based classification and model adaptation to add the testing data into training. Our results outperform other participants given the same set of features, and was selected as the joint winner in KDD Cup 2008.

Keywords
Support Vector Machines, AdaBoost, ensemble method, breast cancer image classification, area under free response receiver operating curves (FROC).

1. INTRODUCTION
The goal of KDD Cup 2008 is to design computational methods for the early detection of breast cancer from X-ray images. The competitors were given a training set consists of 102,294 candidate instances from 118 malignant and 1594 normal patients. Each instance was described by 117 features. There are two challenges in this competition:

Challenge 1: To maximize the area under the free response receiver operating curves (FROC), which is the curve of the precision of patients in the clinically relevant region 0.2-0.3 false positives per image.

Challenge 2: To minimize the false positive rate while maintaining perfect recall.

Several difficulties have been encountered from the distribution of the data as well as the evaluation criterion:

1. The class distribution in the dataset is imbalanced. Only a small fraction of the instances are malignant. In such situation, standard classifiers tend to have a bias in favor of the larger classes and ignore the smaller ones.
2. The evaluation criterion in task 1 is the area under FROC (or AUC). However, the objective function optimized in most of the learning algorithms is the error rate rather than the AUC value.
3. The FROC metric measures the precision per *patient (a patient is represented by a collection of data points)*, while standard classifiers maximizes the precision per *data point*.

Section 2 describes the methodologies applied in our submission. Section 3 discusses some other approaches that are tested with certain level of success but eventually not used in our submission. We conclude and propose some potential future directions in the final section.

2. METHODOLOGIES
Figure 1 provides an overview of our system. We exploit AdaBoost [1], two variations of support vector machine, and a linear classifier trained using the Genetic Algorithm for this task. We train each classifier independently using 10-fold cross validation on the whole training dataset. We also perform an experiment to divide the data into four portions (left/right combining with CC/MLO) and train an independent classifier for each, but found no significant improvement over the accuracy. Final prediction values on testing data points are calculated by combining the outputs of the classifier committee.

The parameters of each classifier are tuned using AUC values as the objective function. Acknowledging the fact that the testing data contains features from only unseen persons, we decide to abandon point-wise cross-validation and adopt patient-wise cross-validation during training. Under this schema the data points of each patient is bundled in a group, and thus forced to be in the same fold in the cross-validation process. We also used stratification technique to ensure that each class is represented with approximately equal proportions in every subset. Although the prediction performance on validation set is much lower in patient-wise CV (≈80% in AUC) than in point-wise CV (>90% in AUC), we believe that the former has better chance to avoid overfitting on the testing data.

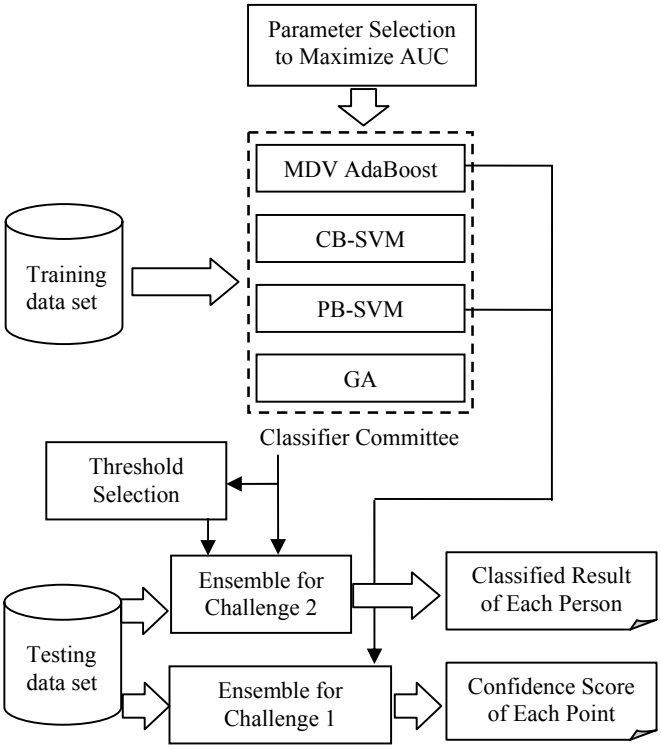

Figure 1: Methodology Overview

The following subsection describes our system in detail: some specific adjustments made in tuning up the SVM for this dataset and evaluation criteria have been discussed first. Then we discuss the AdaBoost (using CART as weak learner) and its performance. Sections 2.3 and 2.4 describe how to combine the predictions of multiple classifiers for challenges 1 and 2, respectively. The corresponding results of each method are listed in table 1 and table 2.

2.1 Classification Using Support Vector Machine

2.1.1 Class-Balanced SVM (CB-SVM)

The major difficulty in the first challenge is due to the imbalance of data since there are 163 times more negative data points than positive points. Consequently, standard classifiers tend to bias in favor of the larger class since by doing so it can reach high classification accuracy. Researchers have proposed several types of solutions to deal with class imbalanced problem such as down-sampling of major class, up-sampling of minor class and adopt class-sensitive loss function [2]. In our submission, we applied a class-sensitive SVM implemented in LIBSVM [3] whose objective is of the form:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} w^T w + C_+ \sum_{y_i=1} \xi_i + C_- \sum_{y_i=-1} \xi_i \\ s.t \quad & y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, i = 1, \ldots, l, \end{aligned}$$

where (x_i, y_i) is an instance-label pair in the training data, Φ is a function that maps input data into a higher dimensional space and C_+, C_- are weights of training errors with respect to the positive and negative examples, respectively. In this task, we set C_+ to be 163 times larger than C_-. The output scores are treated as the confidence values which represent how confident the classifier believes it to be positive.

In the next stage we pursue parameters adjustment. In this particular case, the parameter space consists of training error weight C_- and the Gaussian kernel width γ, while Gaussian kernel function is defined as $K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|_2^2}$. We apply a nested uniform design (UD) procedure [4] to first set out crude search for a highly likely candidate region of global optimum and then confines a finer second-stage search therein. The UD method automatically selects the candidate set of parameter combinations and then carries out a k-fold cross-validation to evaluate the generalization performance of each parameter combination. We chose the area under FROC in the region of 0.2-0.3 false positives as the objective function in the UD method. Once the predictions of all points are generated (in n-fold CV, every point gets exactly one chance to be validated), they will be sent all together to generate the AUC score. This design reaches 77.8% in AUC.

One important observation we have made based on the experimental results of this class-balanced SVM is that it is easier to identify candidates of patients who have more positive instances than patients with fewer positive instances. For example, for patients with more than 10 positive points, the average rank of their highest positive candidate is 960, but for patients with only one positive candidate, the average rank of these positive candidates is 23478. We believe this is because in class-balanced SVM all positive points are treated equally, and therefore the classifier will introduce bias to patients with more positive instances. This is unfortunate since our classifier is evaluated based on the accuracy per person instead of accuracy per instance. This observation motivates us the idea of patient-balanced SVM.

2.1.2 Patient-Balanced SVM (PB-SVM)

To conquer the patient imbalanced problem, we develop a patient-balanced SVM with the following adjustment:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} w^T w + C_i \sum_{i=1}^{l} \xi_i \\ s.t \quad & y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, i = 1, \ldots, l, \end{aligned}$$

where each training data has a corresponding weight C_i. The weights can be determined using two constraints:

$$\begin{aligned} & \sum_i C_i = \sum_j C_j, \forall i,j \\ & s.t\ y_i = 1, y_j = -1. \end{aligned}$$

$$\sum_{i \in patient_m} C_i = \sum_{j \in patient_n} C_j, \forall m,n$$
$$s.t\ y_i = 1, y_j = 1,$$

That is, the summation of weights of all negative examples is equal to the summation of weights of all positive examples; and each individual patient possesses equal summation of positive weights. In this sense, the training error on *rare* (with respect to a patient) positive points will be given higher weights than *common* positive points. The results show that it is 0.8% better than CB-SVM.

2.2 Classification Using AdaBoost

Boosting is a method of finding a highly accurate hypothesis (classification rule) by combining many "weak" hypotheses, in particular when each of which is only moderately accurate.

We tried AdaBoost in our submission as well. The main idea behind AdaBoost is to construct a highly accurate classifier committee by combining many weak learners. The weak learners are only moderately accurate but should be diverse. We choose Real AdaBoost [5] implemented by [6], which supports real-value prediction, and use classification and regression tree (CART) as the weak leaner. The major reason to combine these two is twofold. First CART is inherently suited for imbalanced dataset since its tree is constructed according to the correct classified ratio of positive and negative examples. Second, the model selection procedure can be done simply and efficiently as we can iteratively increase the number of weak learners and stop when the generalization ability on the validation set does not improve. Similar to what PB-SVM does, we apply a modified dependent variable (MDV) AdaBoost [7] to adjust the dependent variable from y_i=+1 to y_i=+1/#positive for positive instances of each malignant patient. The original AdaBoost performs slightly worse than CB-SVM in the training set, but the MDV AdaBoost gets the highest AUC among these four methods. The performance of different parameter setting is shown in Figure 2.We have tested different maximum split numbers (from 1 to 3) in CART and found that the better split number in our task is 2. The patient-based cross validation reveals that it performs the best when the number of weak learners is set to 50.

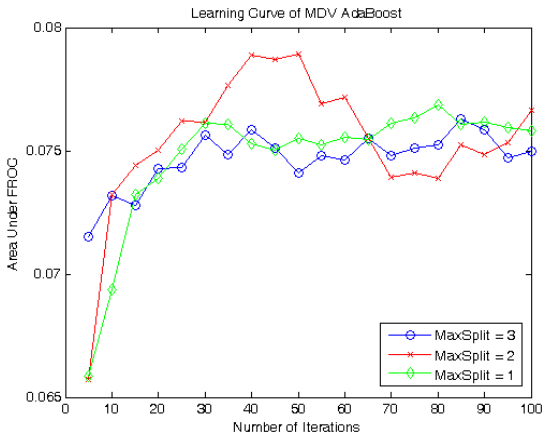

Figure 2: Model Selection for MDV AdaBoost

Table 1. Summary of Performance of Base Classifiers on Training Set and Testing Set

Base Classifier	AUC on Training Set (in %)	AUC on Testng Set (in %)
CB-SVM	77.8	81.3
PB-SVM	78.6	86.7
Original AdaBoost	77.1	80.2
MDV AdaBoost	79.1	87.9

2.3 Combining SVM and AdaBoost for Challenge 1

Table 1 shows results on training and testing set of the four base classifiers. Literature indicates that combining divergent but fairly high performance models into an ensemble can usually lead to a better generalization performance. Table 2 summarizes the results of some plausible ensembles.

The first method we have tried is to train a linear classifier using prediction scores of the three base classifiers together with the original 117 dimensional features. The result is slightly better than the best single classifier (from 79.1% to 79.9%). However, it doesn't generalize well on testing data. We deem that this cascade training method tends to overfit the training data. Second, we simply average the orders (not absolute prediction values) of the three base predictions and inverse the average rank as the prediction. The result goes up a little bit to 80.2%. Third we tried to average the orders of two better classifiers: PB-SVM and MDV AdaBoost and found a notable 2.1% absolute improvement comparing to MDV AdaBoost. This becomes one of our final submissions to KDDCUP 2008. Finally we train a simple weighted order averaging method to combine the four base classifiers. The possible integer weights of each classifier are from 0 to 10. It performs slightly better than the previous method and become another submission. However, it still suffers from the overfitting problem.

Table 2. Comparison of Different Ensemble Methods on Training Set and Testing Set

Ensemble Method	AUC on Training Set (in %)	AUC on Testng Set (in %)
Linear Classifier	79.9	85.3
Avg. of Three Classifiers	80.2	89.0
Avg. of Two Best Classifiers	81.2	89.5
Weighted Avg. of All Base Classifiers	81.8	87.6

2.4 Methods for Challenge 2

The aim of challenge 2 is to increase the precision while ensuring perfect sensitivity (i.e. no false negative). To tackle this challenge, we first generate predictions from four different classifiers: AdaBoost, CV-SVM, PB-SVM, and the Genetic Algorithm (GA). In GA, we use the area under FROC directly as the fitness function. The experiments show that GA does better in

recognizing patients that have fewer positive points. Using the prediction scores of a classifier, we first rank the patients by comparing their top-ranked candidates, and then give one vote to the top x% of the patients. Since there are four classifiers, a patient will receive at most four votes. Eventually we pick the patients who receive at least y votes as the positive ones and the rest as negative ones. The results of 10-fold cross-validation show that when we set x=80% and y=2, it can reach 100% sensitivity in every fold with better precision (23%).

3. DISCUSSION

This section discusses some other ideas that we have implemented for this challenge. The experimental results reveal that they might not be as good as the previously discussed ones but still possess their own value and could be applicable to other cases.

3.1 Location-based Clustering and Classification

Another observation we have made is that the image data might possess features reflecting their location information on the chest, which might have nothing to do with its cancer status (i.e. whether it is malignant or not). To eliminate the interference from these features, we plan to first cluster the data points using their X-Y coordination (i.e. the relative coordination from center) and train independent classifiers in each cluster. In this sense, a data point is compared with only points in the same region. Therefore, when a test data comes in, our system first determines which cluster it belongs to, and applies the corresponding classifier on it. To test this idea, we tried to use K-means first to cluster points into 20 groups based on their location, and use them to train 40 different classifiers (it is twice as many since we distinguish CC from MLO data points). The experimental results reach roughly 70% in AUC. The reason why it is not as competitive is due to the fact that dividing data into smaller groups makes the already sparse positive data even sparser, which inevitably lower the classification performance.

3.2 Model Adaptation Using the Testing Data

According to the experiment we have described, there is a large performance gap between point-wise CV (>90%) and patient-wise CV (80%). This reveals the fact that a classifier can perform much better if it sees some instances of a patient in training. Based on this intuition, we design a method to incorporate highly confident testing data points to *adapt* the trained model. First we train a classifier using training data and generate predictions on testing data. Then we add the highly confident testing data into the training set to train an *adapted classifier*. Finally this adapted classifier is used to generate the final prediction on the testing data. The preliminary experiments we have pursued using manually selected threshold shows limited improvement. However, we would like to perform a more thorough experiment (including automatic choosing of threshold) on the testing data of KDDCUP 2008 after the announcement of testing labels.

3.3 Exploiting Patient ID Feature

As reported by Perlich et al. [8], the patient ID (PID) in the dataset reveals powerful predictability to class label and can significantly improve the results. Here we would like to test how well our model performs while such leakage information is added as an additional feature. The PID is discretized in the same manner as suggested in [8]. The AUC on testing data using MDV AdaBoost reaches 94.69%, which is 1.4% higher than the best results reported in [8]. This experiment confirms that such leakage information on patients is very important, and our model still outperforms other models when this feature is applied.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a solution for imbalanced data classification that aims at optimizing the area under FROC for KDD Cup 2008 which addresses three challenging issues: dealing with imbalanced class distribution, optimizing area under FROC criterion and maximizing precision per patient. We propose to integrate patient-balanced and class-balanced SVMs (both of which select parameters using a nested uniform design procedure) with modified dependent variable AdaBoost for challenge 1. We apply another ensemble system which combines the above three methods plus GA to predict the labels of patients for challenge 2. The parameters for challenge 2 are adjusted to maintain 100% sensitivity and optimize the precision. The results for both challenges are promising, though (in particular for challenge 2) still have room for improvement.

There are several plausible future directions. First, we would like to investigate how the domain knowledge (e.g. the alignment of CC and MLO data or the other medical information about the patient) can be incorporated into the system. Second, we would like to add several other classifiers (e.g. CRF, maximum entropy model, etc.) into the ensemble. Third, we are interested in exploring better ensemble methods for integration.

5. REFERENCES

[1] Y. Freund and R. 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.

[2] G. M. Weiss. Mining with rarity: a unifying framework. ACM SIGKDD Explorations Newsletter. Vol. 6, 7-19, 2004.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[4] C.-M. Huang, Y.-J. Lee, D. K. J. Lin and S.-Y. Huang. Model Selection for Support Vector Machines via Uniform Design, A special issue on Machine Learning and Robust Data Mining of Computational Statistics and Data Analysis. Vol. 52, 335-346, 2007.

[5] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. Machine Learning, 37(3):297-336, 1999.

[6] GML AdaBoost Matlab Toolbox. Software available at http://research.graphicon.ru/machine-learning/gml-adaboost-matlab-toolbox-6.html

[7] R. M. Bell, P. G. Haffner and C. Volinsky, J. Modifying boosted trees to improve performance on task 1 of the 2006 KDD challenge cup. ACM SIGKDD Explorations Newsletter. Vol. 2, 47-52, 2000.

[8] C. Perlich, P. Melville, Y. Liu, G. Swirszcz, R. Lawrence and S. Rosset. Winner's Report: KDD CUP Breast Cancer Identification. In Proceedings of the KDD-08 Workshop on Mining Medical Data, 2008