SCIENCE CHINA
Information Sciences

• RESEARCH PAPERS •

July 2010 Vol. 53 No. 7: 1369–1378
doi: 10.1007/s11432-010-3107-9

Detecting malignant patients via modified boosted tree

DONG CaiLing[1], YIN YiLong[1]* & YANG XiuKun[2]

[1]*School of Computer Science and Technology, Shandong University, Jinan* 250101*, China;*
[2]*College of Information and Communication, Harbin Engineering University, Harbin* 150001*, China*

Received June 9, 2009; accepted December 23, 2009

Abstract As one of the most effective measures to extract useful information from medical database and provide scientific decision-making for diagnosis and treatment of diseases, medical data mining has become an increasingly hot topic in the last few years. Some of the intrinsic characteristics of medical databases, such as the huge volume and imbalanced samples as well as stringent performance standards, make this mining process particularly challenging. By elaborating various challenges existing in Task 1 of KDD Cup 2008 competition, this paper analyzes some potential solutions to these problems and presents a modified boosted tree as the final classification model. This model ranked the fourth among all the solutions to Task 1. We hope that our analysis and solutions to these challenges would contribute to the development of medical data mining applications.

Keywords classification, boosted tree, multiple instances scoring, Real AdaBoost, breast cancer detection, medical data mining

Citation Dong C L, Yin Y L, Yang X K. Detecting malignant patients via modified boosted tree. Sci China Inf Sci, 2010, 53: 1369–1378, doi: 10.1007/s11432-010-3107-9

1 Introduction

Modern health care process generates huge amounts of clinical data. Evaluation of such data may lead to the discovery of hidden patterns which could significantly enhance our understanding of disease propagation and management. Aimed at extracting useful information from medical database and providing scientific decision-making for diagnosis and treatment of diseases, medical data mining has become a hot topic in the last few years.

Distinct from other data mining applications, medical data mining is characterized by its specific clinical dataset as well as stringent performance standards. As elaborated by Cios and Moore [1], vast quantity of medical data usually comes from different sources. Such heterogeneous data could result in poor mathematical description. In addition, sensitivity/specificity measures which are widely used in error analysis can hardly ensure the accuracy in disease prediction. Besides, bearing upon ethical issues and privacy, little descriptive information is explicitly provided for clinical data. All of these characteristics make the process of medical data mining extremely challenging.

As one of the important goals of medical data mining, early detection and prevention is essential in fighting diseases. With the aid of rapid-computation and accurate measurement of advanced computers,

*Corresponding author (email: ylyin@sdu.edu.cn)

CAD (computer-aided detection) that makes full use of digital image processing techniques and data mining principles has been one of the most promising adaptations of computer-prompting technologies in clinical medicine.

KDD Cup 2008 competition[1)] focuses on computer-aided detection in screening mammography, which is of vital importance since breast cancer is still the second leading cause of cancer deaths for women today although the incidence decreased by 2.2% per year from 1999 to 2005 due to the early detection [2]. In the following, we will take this computation as a case to study various challenges existing in medical data mining. We analyze some potential solutions to these challenges and construct a final classification model to achieve good prediction results on breast cancer.

The remaining part of the paper is organized as follows: Section 2 gives a brief description to this competition and some characteristics. Section 3 presents our solutions to addressing the challenges originating from the complexity of dataset and evaluation criterion in this competition. Section 4 elaborates the whole process of constructing final classification model with details of modifications on decision tree and AdaBoost as well as selection of parameters. Finally, we give the results and conclusions in section 5 and section 6, respectively.

2 Competition description

In KDD Cup 2008 competition, four X-ray images are provided for every patient with each breast imaged from two different directions. Universal paradigm of addressing CAD problem is mainly composed of four stages: candidate generation, feature extraction, classification and visualization [3, 4]. Unlike Bi and Liang [3] who worked on all the four stages of CAD to detect pulmonary embolism via exploiting spatial vascular structure, the first two stages concerning image processing in this competition have been done beforehand. We just focus on classification and try to build an efficient classifier to differentiate malignant cancers from the benign ones. In the datasets, each candidate ROI (region of interest) is described by a vector of 117 numerical features which are extracted from those images. There are 102294 candidates in the training dataset, including 118 patients with malignant cells among the total 1712 patients. In addition, some descriptive features such as image ID, lesion ID, class label, etc. are provided accordingly. Another 94730 candidates in the same format without class label or lesion-ID are given as test dataset.

The goal of Task 1 of KDD Cup 2008 is to produce a confidence score for each candidate in test dataset indicating its likelihood of being cancerous. Moreover, FROC (free-response receiver operating characteristic) curve is chosen as the evaluation criterion, which encourages competitors to detect as many true positive patients as possible in the clinically relevant region of 0.2–0.3 FPs (false positives) per image.

The following peculiarities of the dataset and evaluation criterion for this task demonstrate several typical characteristics of medical data mining:

- Huge datasets with high dimensionality. As mentioned before, the training dataset and test dataset are huge matrices of 102294 × 117 and 94730 × 117 respectively.

- The semantics of features is unknown. In both training and testing datasets, each candidate is described by 117 numerical features which are normalized to a unit range, but no semantic explanation is provided due to privacy issue. It imposes more difficulties on understanding the dataset and exploiting spatial correlation among these features.

- Imbalanced dataset. Datasets are severely imbalanced between positive and negative classes. In the training dataset, positive candidates only account for about 0.6%.

- Multiple instances scoring. Multiple candidates may belong to the same lesion and multiple lesions may belong to the same image. Besides, each patient is described by four images. Such situation produces several many-to-one mapping relationships, resulting in the problem of multiple instances scoring.

- Strict evaluation criterion. FROC curve with confidence interval for the expected number of false positive markings per image at a given sensitivity has been proved to be an efficient evaluation criterion

[1)] http://www.kddcup2008.com/

in CAD systems [5]. Values on Y-axis and X-axis of the FROC curve here are of different levels: Y-axis represents sensitivity of the true positive patients, while X-axis represents average FP rate of candidates per image. Hard constraints are imposed on maximal patient sensitivity, which limits FPs to the range of [0.2, 0.3].

3 Analysis of problems in datasets

3.1 Feature selection

Due to the heterogeneity of medical datasets, it is inevitable that many redundancies exist among these high-dimension data. Selecting a set of representative features which capture the most informative properties can not only improve the efficiency of the classification models but also decrease the time and spatial complexity. Feature selection algorithms mainly fall into two categories according to whether or not this process depends on learning algorithms. The filter model focuses on characteristics of training instances, which is done independently of learning algorithms, while the wrapper model aims to find representative features suitable for a predefined learning algorithm [6, 7]. While wrapper model can better capture the features with regard to a specific classifier, filter model is more efficient in computation. Due to the high dimensionality of our dataset and coordinate relation among features, we decided to use filter model to select features without too much computational cost.

We tried two feature selection approaches under Weka [6] workbench on the training dataset. Firstly, we gradually chose features with the best classifying performance using BestFirst (Forward) and evaluated their predictive ability by CfsSubsetEval. Using 10-fold cross-validation on this approach, we obtained only 17 features. Such reduction of feature sets denotes the close correlation among these features. Since we planned to use tree as base classifier, in the second approach, we ranked all the features based on their information gain with respect to classes. That is, we chose Ranker as search method and InfoGainAttributeEval as attribute evaluator. (Since this approach cannot handle continuous values, we discretized all the feature values into different intervals according to their distributions on Weka beforehand). The maximal average merit of the results acquired from 10 fold cross-validation is only 0.011 +−0. Among the 20 features with average merit between 0.005 and 0.011, 70% of these features are the same as those we obtained from the first approach.

We defined the following four groups of features as the candidate feature sets:

- Features_a: feature set generated by BestFirst (Forward) + CfsSubsetEval.
- Features_b1: feature set generated by Ranker+ InfoGainAttributeEval with average merit between 0.002–0.011.
- Feature_b2: feature set generated by Ranker+ InfoGainAttributeEval with average merit between 0.003–0.011.
- Feature_all: original feature set without feature selection.

Then we evaluated the representativeness of these feature sets by classifying the training dataset using tree J48 on Weka. Here we directly focus on the prediction on candidates rather than consider its capability of detecting patients. The results based on 5-fold cross-validation including the number and percentage of TP (True Positive), FP (False Positive), TN (True Negative) and FN (False Negative) are shown in Table 1. Surprisingly, Featues_a performed quite well with such a small feature subset while other feature sets performed only slightly better, which to some extent substantiates the representativeness of these features. All of the feature sets produced good accuracies on negative candidates but bad detection rates on positive candidates because of the severe imbalance between the two classes.

Since the prediction accuracy of positive candidates is more important, we choose the 60 features in Feature_b2 with higher TP value as the final feature set of our model.

Constructing proper wrapper models could also achieve good results. For instance, Puuronen et al. [8] proved the efficiency of a local feature selection approach with dynamic integration around decision trees. Specially, faced with the similar problem of high dimensionality, instead of selecting representative features, Bell et al. [9] kept all the original features and added 5 more informative features constructed

Table 1 Performance of each extracted feature set

Feature sets	Result				Size of feature sets
	TP	FP	TN	FN	
Featues_a	95 (15.25%)	528 (84.75%)	101650 (99.98%)	21 (0.02%)	17
Featurs_b1	107 (17.17%)	516 (82.83%)	101632 (99.96%)	39 (0.04%)	86
Feature_b2	**159 (25.52%)**	**464 (74.48%)**	101569 (99.90%)	102 (0.10%)	**60**
Feature_all	105 (16.85%)	518 (83.15%)	101639 (99.97%)	32 (0.03%)	117

from five different kinds of neighborhood for each candidate. These 5 features showed their distinguished classification capability, which confirmed Bi and Liang's strategy of deeply exploration on spatial correlations in medical data mining [3].

3.2 Imbalanced datasets

Many real world applications suffer from class imbalance problem since normal instances are usually abundant while the anomalous instances in which we are interested are relatively scarce. Standard machine learning algorithms searching for accuracy on the whole will extremely skew class boundary to positive class and thus result in high FPs, which is exactly the biggest challenge in clinical prediction.

To date, there have been many techniques tackling this problem, which can be mainly divided into three categories. The first one is to change the class distribution by re-sampling original dataset [10–16]: under-sampling the majority class or over-sampling the minority class. Although under-sampling has been regarded as a good method of increasing the sensitivity of classifier, it may discard some potentially useful data which could be important for classification. On the other hand, over-sampling tends to incur the over-fitting problem due to the fact that it needs to repeatedly take samples from the original dataset. Methods such as SMOTE [17] can alleviate this problem by creating synthetic positive instances instead of using repeated ones. However, it still unavoidably induces some noises into datasets. To overcome these deficiencies, some other methods are created: Liu et al. [15] proposed an ensemble output approach (EasyEnsemble) as well as a cascaded training approach (BalanceCascade) to efficiently utilize the instances in majority class neglected by under-sampling; Kubat et al. [11] also managed to find representative instances from majority class by removing "borderline" and "noisy" instances. The second category of approaches is to define fixed and unequal misclassification costs between classes [10]. That is, try to accommodate class-imbalance and cost-sensitive problems by exploiting their intrinsic connections [10, 12, 13, 18]. Variants of AdaBoost are representative in this category and they all focus on increasing weights of those instances with higher misclassification cost [15, 19]. Besides, low false positive is always needed in cost-sensitive problems, thus it becomes one of the focal points dealing with imbalanced dataset [3, 20]. The third kind of approaches suggested for class imbalance problem is to find a set of more representative and balanced data from the original imbalanced dataset rather than create one. A good example is provided by Ertekin et al. [16], who take advantage of active learning and efficiently find an informative subset with relatively smaller imbalance ratio within the margin of SVM (support vector machine).

In this task, the multi-instance property indicates that different number of candidates may belong to different patients, which also cover different lesions. Such a situation makes it hard to perform sampling on this dataset: under-sampling may inevitably delete some candidates with dominant features influencing the final classification results, while noises generated by over-sampling could make the process of detection more complicated. Thus we tackle the problem of class imbalance in terms of cost-sensitivity. That is, we keep the original dataset and address the class imbalance problem by exploiting the intrinsic properties of AdaBoost, i.e., giving more penalties to those misclassified instances during the training process, and thereby decreasing FPs.

3.3 Multiple instances scoring

Multi-instance problem is quite common in medical dataset since instances used to describe patients are usually extracted from images. Besides, the number of lesions is always different with regard to specific patients. In multi-instance learning, instances in training dataset belong to several "bags". A bag is labeled as positive if it contains at least one positive instance, otherwise it will be labeled as negative. The trick of multi-instance learning is that we have to correctly identify each positive instance in each "bag" while the credit gains nothing even if we find more than one positive instance for the same "bag" [21, 22].

In this task, it would have been a complicated procedure if we had taken into accounts all of the many-to-one relationships among candidates, lesions, images and patients. Moreover, the lack of lesion ID for test dataset makes these correlations more obscure. Since the final evaluation criterion of the task is based on patient level and candidate level, to cater for the measurement criterion we just directly apply the many-to-one mapping relationship between candidates and patients.

4 Classification model—modified boosted tree

All the above characteristics of this medical dataset make it impossible for us to apply an off-the-shelf classifier. Partially inspired by the strategy proposed by Bell et al. [9] in classifying pulmonary embolism, we choose boosted tree as our final classification model due to the state-of-the-art innovations on trees and AdaBoost:

Trees are competitive among many classifiers in multiple aspects. First of all, they can perform well on large data with many features while little prior information about their correlations is needed [23]. Secondly, they are able to handle both numerical and categorical data. These two advantages substantiate our approach in utilizing the huge dataset with continuous feature values to some extent. Furthermore, there have been some researches on using trees to handle multi-instance problems [22, 24, 25], which give us some hints in addressing this task.

As a meta-algorithm, AdaBoost can be used in conjunction with many other learning algorithms. It constructs a "strong" classifier as a set of combination of some "simple" "weak" classifiers by means of focusing on and retraining those instances misclassified by previous classifiers. Repeatedly training on misclassified instances can improve the overall accuracy and decrease FPs. Another main superiority of AdaBoost is its good generalization on a number of predictive variables with a large margin and its robustness to the overfitting issue [23, 26]. Moreover, Real AdaBoost [27], as a confident-related AdaBoost, produces a series of scores at each boosting step, which can be directly used to sort the confidence of candidates in our classification procedure. On the other hand, combination of AdaBoost and trees has been efficiently used in many applications such as spam filtering and face detection, which shows its superiority on dealing with such tasks that demand low FPs [23].

Some modifications have to be made on boosted tree to address the idiosyncrasies of dataset and scoring criterion. Thus we developed a classification model named "modified boosted tree", using modified decision tree as the base classifier and Real AdaBoost as the wrapper.

4.1 Base classifier: modified decision tree

Two problems should be taken into account in our classification procedure on decision tree: the partition criterion of selecting the best splitter for each node and the final output of confidence.

4.1.1 *Partition criterion*

Finding appropriate attributes that best split the given instances as tree nodes is the core of constructing decision trees. Trees do a good job on discrete attributes, so numerical attributes are often mapped into discrete intervals through different algorithms. Traditionally, one firstly just considers the adjacent instances that differ in their class labels in a sorted instance set according to each numerical attribute, then evaluates these candidates by computing information gain, and finally chooses the best one [6, 28, 29].

Due to the fact that our task is based on a severely imbalanced dataset, the candidate thresholds would have been relatively few compared with the whole dataset if we had used the traditional method.

As an improvement, for each attribute, we decided to consider all the adjacent instances with different attribute values in the increasingly sorted instance set, and choose the medians as the candidate thresholds, regardless of their class labels. The details of our partition criterion are shown in Algorithm 1. The information gain we use here is based on entropy.

```
Algorithm 1: Partition criterion for constructing decision tree
Input: A matrix of training dataset $m \times n$, where $m$, $n$ represent the number of instances and attributes,
       respectively.
Define:
  $x_j \leftarrow$ attribute value of the $j$th instance, where $0 \leqslant j \leqslant m-1$
  $G_i \leftarrow$ information gain with regard to an arbitrary value $i$
  Attributes set $\leftarrow$ a set of attributes constituted by the whole attributes
Process:
  foreach attribute $A$ in Attributes set do
    $V \leftarrow$ The value set of attribute $A$ corresponding to all the instances
    Sorted-V $\leftarrow$ an ordered list by sorting $V$ in increasing order
    foreach value pair $(x_j, x_{j+1})$ in Sorted-V do
      if $x_j \neq x_{j+1}$ then
          Set $t_j = (x_j + x_{j+1})/2$ be the candidate threshold and calculate $G_{t_j}$
      splitter$_A \leftarrow$ candidate threshold $t_j$ of $A$ with biggest $G_{t_j}$
    Splitters $\leftarrow$ add splitter$_A$
Output:
  Best-Splitter $\leftarrow$ the candidate threshold $t_j$ in Splitters with biggest $G_{t_j}$
```

In this way we are able to perform a more thorough search for candidate thresholds, which definitely increases the time and spatial complexities but it may provide more potential to find the "best" splitters to construct good trees.

4.1.2 *Output of confidence*

Our task is actually a soft classification problem since the final goal is to return a probability instead of a hard label for each candidate. For this classification tree, we define the confidence of each leaf node as the probability of positive instances arriving at this node. So our binary tree finally produces real-valued predictions in the range of [0, 1] for each instance.

4.2 Wrapper: modified Real AdaBoost

We use Real AdaBoost instead of AdaBoost to combine the real-valued predictions produced by our tree in each iteration step. That is, at each step $T (0 \leqslant t \leqslant T)$ in total T training rounds, our modified tree predicts a confidence value $h_t(x_i)$ for each candidate x_i $(0 \leqslant i \leqslant n)$, ($n$ is the total number of candidates in the dataset). Then Real AdaBoost aggregates T values together as the final confidence score of x_i: $H(x_i) = \sum_1^T \alpha_t h_t(x_i)$.

4.2.1 *Selection of α_t in Real AdaBoost*

Parameter α_t in AdaBoost determines the weight of weak classifier and the choice of its value depends on the types of different weak classifiers [23]. Schapire et al. [27] pointed out that greedily minimizing Z_t (a normalization factor in AdaBoost) on each round of boosting is a reasonable approach to minimizing training error. As a variable in Z_t, the decrease of α_t actually benefits the whole accuracy of the training set. Furthermore, small values of α_t are capable of keeping base classifiers from learning too much from any of the base classifiers [9]. We eventually assign a constant small value 0.1 to α_t rather than change it dynamically in traditional AdaBoost.

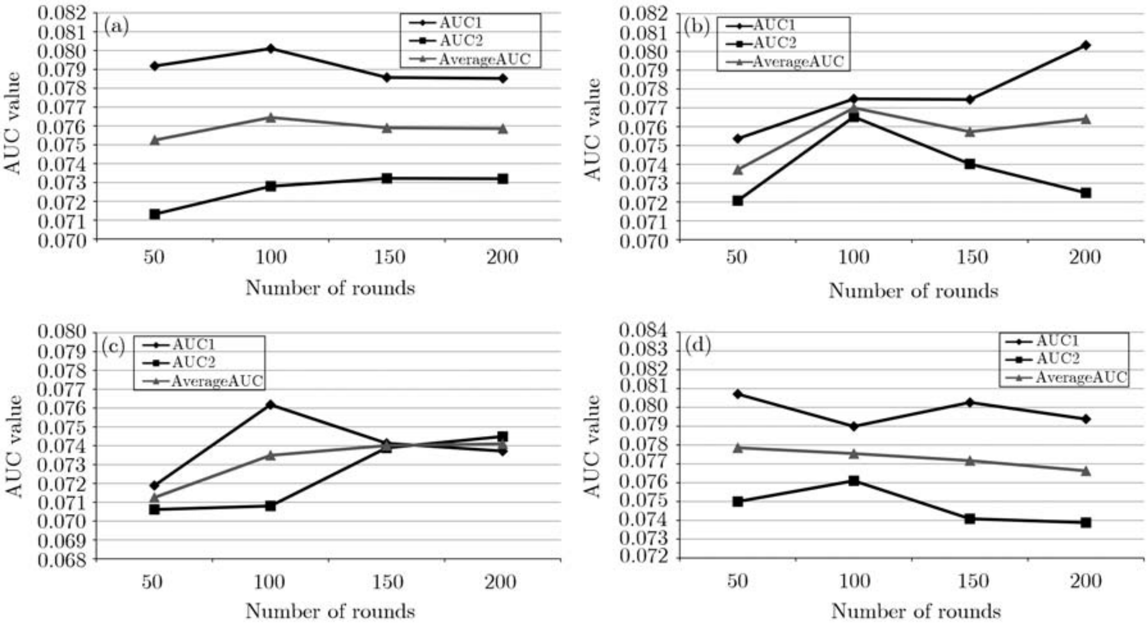

Figure 1 AUC values based on different number of nodes and training rounds. (a) Number of nodes=64; (b) number of nodes=128; (c) number of nodes=192; (d) number of nodes=256.

4.2.2 *Endowing patients with equal chances to be identified*

Assuming "evidences" denotes the number of positive candidates belonging to a malignant patient, it is obvious that those patients with lower evidences have fewer chances to be identified and therefore they are more easily misclassified. In addition, as a result of the strict limitation on FPs, even small changes in labeling of candidates can yield dramatic changes in the final result. So it is advisable to pay more attention to patients with low evidences.

For AdaBoost, Schapire et al. [27] define the margin of a labeled instance (x, y) to be $y \times f(x)$ and indicate that larger margins imply lower generalization error. In addition, they also regard $\mid y \times f(x) \mid$ as a measure of the confidence of hypothesis H's prediction. Likewise, Friedman et al. [30] suggested that the magnitude of hard label y_i, in a sense, serves as the weight of instance i in boosting.

Based on these logical proofs and enlightened by Bell et al. [9], we feed evidences into the dependent variable of positive candidates and therefore focus on the patients with low evidences. That is, we replace $y_i = +1$ with $y_i = +1/\text{evidences}$ for each positive patient and provide them with the same weight.

4.3 Selection on tree size and training rounds

Tree size and training rounds are two important parameters for the whole classification model. It is inevitable that decision tree will suffer from over-fitting if it grows deep enough to correctly classify each instance [30]. Freund and Schapire [31] also pointed out that AdaBoost is more likely to over-fit if it runs too many rounds. Thus, we applied some small values on these two parameters. The number of nodes is 64, 128, 192 or 256 separately, while the number of iterative rounds is 50, 100, 150 and 200 respectively.

In this experiment, we randomly divided the original training dataset into two subsets: one is for training and the other is for testing, with the special guarantee that each subset includes exactly half of the positive patients. Final decision was made according to the average value of the two AUC (area under the curve) values derived from 2-fold cross-validation.

According to the visualized results shown in Figure 1, we may come to a conclusion that our tree with more than 256 nodes may induce over-fitting since the corresponding AUC values with parameter of "Number of nodes = 256" decline constantly as the number of rounds increase. Better performance can be achieved by setting "Number of nodes = 192" with "Number of rounds = 150", since its AUC values are relatively stable and three AUCs converge to this point. Thus we eventually constructed our modified boosted tree with 192 nodes and make it iterate 150 times with Real AdaBoost.

4.4 Selection of cut point

In order to evaluate the performance of our modified boosted tree and most importantly, to evaluate the prediction capability, we have to find an appropriate cut point to map confidence values to hard labels. This threshold makes vital impact on classification variance due to the stringency of the evaluation criterion. It could be a theoretically accurate approach to determining such value via estimating FP probability for each candidate using logistic regression [9], but for our huge dataset, to ensure a qualified prediction with adequate accuracy, we prefer to use an on-line checking scheme. The false positive rate is checked whenever an instance is classified to make sure the average FP rate per image is less than 0.3. The details of this procedure are shown in Algorithm 2. As outputs of this algorithm, Split-Point is the cut point we finally choose and the sensitivity refers to the sensitivity of patients we are hunting for.

```
Algorithm 2: Selection of cut point for performance evaluation
Input: $n$ candidates with corresponding confidences (0.0–1.0), original label ($-1$, 1), imageID, and patientID
Define:
  Sorted-Set: candidate set sorted by confidences in decreasing order
  Fa-Num: number of false positive patients
  image-Num: number of images in this data set
  Detected-ID: ID set of all the detected true positive patients
  Split-Point: the cut point
  Result-label: hard label ($-1$, $+1$) classified based on Split-Point
  Sensitivity: the proportion of true positive patients
  Attributes set ← a set of attributes constituted by the whole attributes
Process:
  for $i = 1, 2, \ldots, n$ do
    if Sorted-Set[$i$].original label = $-1$ then
        Fa-Num++
        if Fa-Num/image-Num> 0.3 then
          Split−Point=$i$
          break
    else
        Detected-ID← add Sorted-Set[$i$].patient ID
        Result−label[$i$] = +1
  for $j$=Split-Point to $n$ do
        Result−label[$j$]=−1
Output:
  Split-Point
  Sensitivity ← distinct ID in Detected-ID/number of patients
```

5 Final results

Our prediction on test dataset produced by modified boosted tree finally ranked the fourth in Task 1 of KDD Cup 2008 with AUC =0.0881. The top three teams and their results on Task 1 are shown in Table 2 [4][2].

The winning team achieved such a high score by exploiting a leakage in the dataset: they discovered a statistical relationship between patient ID and the likelihood of malignancy of the patient implied in the dataset, which may be caused unintentionally in the process of data preparation in this competition [32]. Although it is an unrealistic approach in cancer prediction or medical data mining, they disclose a risky phenomenon in modeling competitions and finally got the best predictive results by utilizing a bagging

[2] The results are publicly announced in http://kddcup2008.com/

Table 2 Top 3 teams in Task 1 of KDD Cup 2008

Team /Affiliation	AUC value
Predictive Modeling Group, IBM Research	0.0933
National Taiwan University	0.0895
Wayne State University	0.0894

linear SVM as classifier. Lo et al. [33] won the second place with an ensemble model. They modified a class-sensitive SVM (PB-SVM) to tackle the patient imbalanced problem, and combined its predictions with the results generated by MDV AdaBoost [9]. The average value of the orders obtained from these two classifiers is used as the final confidence of each candidate. In their model, they also considered the influence of the number of positive candidates with regard to each patient and adjusted the weight of dependent variables in the same way as we did in subsection 4.2.2. Besides, the fact that better predictions are achieved by MDV AdaBoost than other basic classifiers confirms its effectiveness again and moderately implies the potential availability of our model. The result of the third place is so close to the second place, but unfortunately their strategy is still unknown because no publication or report related to it is available.

Although no detailed information about the strategies used by the winning teams on data preprocessing, model construction has been provided, their success uncovers the importance of thoroughly investigative data analysis. This process has always been overwhelmed by other distinct challenges but in fact is the core of data mining applications especially for medical data mining.

6 Conclusions

Automated analysis on medical data and clinical trials is becoming increasingly important in health care. Our competition experience substantiates the fact that comprehensive understanding and intensive analysis on dataset is the core of medical data mining. In general, medical datasets are voluminous and high dimensional. Some features are represented by coordinate values in medical images, which are inevitably correlated. Furthermore, semantic explanations for features are seldom provided for the sake of privacy protection, exerting more pressure on feature selection and feature extraction. Class imbalance is also one of the typical features of medical datasets, which makes some "off-the-shelf" algorithms and traditional accuracy evaluation methods inefficient. Since cost sensitivity is closely related to class imbalance, clinical detection usually imposes stringent restriction on the rate of false positives, which is the most challenging process due to its decisive influence on evaluating classifiers and therefore becomes the keystone of medical data mining.

Development of cutting-edge data mining technology is of vital importance to clinical detection. More work needs to be done on data-preprocessing, model construction and performance evaluation in the future to better address emerging challenges.

Acknowledgements

We would like to thank Chuancong Gao, Maria Biryukov and Qin Ma for their helpful comments and constructive advice on structuring the paper.

References

1 Cios K J, Moore W. Uniqueness of medical data mining. Artif Intel Med, 2002, 26: 1–24

2 American Cancer Society. Breast Cancer Facts & Figures, http://www.cancer.org/, 2009

3 Bi J B, Liang J M. Multiple instance learning of pulmonary embolism detection with geodesic distance along vascular structure. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Anchorage, Alaska: IEEE Computer Society, 2008. 1–8

4 Rao R B, Yakhnenko O, Krishnapuram B. KDD cup 2008 and the workshop on mining medical data. ACM SIGKDD Explor, 2008, 10: 34–38

5 Bornefalk H, Hermansson A B. On the comparison of FROC curves in mammography CAD systems. Med Phys, 2005, 32: 412–417

6 Witten I H, Frank E. Data Mining: Practical Machine Learning Tools and Techniques. 2nd ed. San Francisco, CA: Morgan Kaufmann, 2005

7 Das S. Filters, wrappers and a boosting-based hybrid for feature selection. In: Proceedings of the Eighteenth International Conference on Machine Learning. San Francisco: Morgan Kaufmann, 2001. 74–81

8 Puuronen S, Tsymbal A, Skrypnyk I. Advanced local feature selection in medical diagnostics. In: 13th IEEE Symposium on Computer-Based Medical Systems. Washington DC: IEEE Computer Society, 2000. 25

9 Bell R M, Haffner P G, Volinsky C. Modifying boosted trees to improve performance on task 1 of the 2006 KDD challenge cup. ACM SIGKDD Explor Newslett, 2006, 8: 47–52

10 Karagiannopoulos M, Anyfantis D, Kotsiantis S B, et al, A wrapper for reweighting training instances for handling imbalanced data sets. In: Proceedings of the 4th IFIP International Federation for Information Processing. Boston: Springer, 2007. 247: 29–36

11 Kubat M, Matwin S. Addressing the curse of imbalanced training sets: one-sided selection. In: Proceedings of the 14th International Conference on Machine Learning. Tennessee: Morgan Kaufmann, 1997. 179–186

12 Japkowicz N. The class imbalance problem: significance and strategies. In: Proceedings of the 2000 International Conference on Artificial Intelligence, 2000. 111–117

13 Jo T, Japkowicz N. Class imbalances versus small disjuncts. ACM SIGKDD Explor Newslett, 2004, 6: 40–49

14 Weiss G M. Mining with rarity: a unifying framework. ACM SIGKDD Explor, 2004, 6: 7–19

15 Liu X Y, Wu J X, Zhou Z H. Exploratory under-sampling for class-imbalance learning. In: Proceedings of the 6th IEEE International Conference on Data Mining. Washington, DC: IEEE Computer Society, 2006. 965–969

16 Ertekin S, Huang J, Bottou L, et al. Learning on the border: active learning in imbalanced data classification. In: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management. New York: ACM, 2007. 127–136

17 Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique. J Artif Intel Res, 2002, 16: 321–357

18 Domingos P. MetaCost: a general method for making classifiers cost-sensitive. In: Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining. San Diego: ACM, 1999. 155–164

19 Ting K M. An empirical study of MetaCost using Boosting Algorithms. In: Proceedings of the Eleventh European Conference on Machine Learning. Berlin: Springer, 2000. 413–425

20 Wu S H, Lin K P, Chen C M, et al. Asymmetric support vector machines: low false-positive learning under the user tolerance. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2008. 749–757

21 Dietterich T G, Lathrop R H, Perez L T. Solving the multiple-instance problem with axis-parallel rectangles. Artif Intel, 1997, 89: 31–71

22 Zhou Z H. Multi-instance learning from supervised view. J Comput Sci Tech, 2006, 21: 800–809

23 Carreras X, Márquez L. Boosting trees for anti-spam email filtering. In: Proceedings of the 4th International Conference on Recent Advances in Natural Language Processing, Tzigov Chark, Bulgaria. 2001. 58–64

24 Blockeel H, Page D, Srinivasan A. Multi-instance tree learning. In: Proceedings of the 22nd International Conference on Machine Learning. New York: ACM, 2005. 57–64

25 Chevaleyre Y, Zucker J D. Solving multiple-instance and multiple-part learning problems with decision trees and decision rules. Application to the mutagenesis problem. In: Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence. Berlin: Springer, 2001. 204–214

26 Rätsch G, Onoda T, Müller K R. Soft margins for AdaBoost. Mach Learn, 2000, 42: 287–320

27 Schapire R E, Singer Y. Improved boosting algorithms using confidence-rated predictions. Mach Learn, 1999, 37: 297–336

28 Mitchell T. Machine Learning. New York: McGraw-Hill, 1997

29 Han J W, Kamber M. Data Mining: Concepts and Techniques. 2nd ed. San Francisco: Morgan Kaufmann, 2006

30 Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting. Annals of Statistics, 1998, 28: 337–407

31 Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting. In: Proceedings of the Second European Conference on Computational Learning Theory. Berlin: Springer, 1995, 55: 23–37

32 Perlich C, Melville P, Liu Y, et al. Winner's Report: KDD CUP Breast Cancer Identification. ACM SIGKDD Explor, 2008, 10: 39–42

33 Lo H Y, Chang C M, Chiang T H, et al. Learning to improve area-under-FROC for imbalanced medical data classification using an ensemble method. ACM SIGKDD Explor, 2008, 10: 43–46