Ensemble Learning with Active Example Selection for Imbalanced Biomedical Data Classification

Sangyoon Oh, Min Su Lee, and Byoung-Tak Zhang

Abstract—In biomedical data, the imbalanced data problem occurs frequently and causes poor prediction performance for minority classes. It is because the trained classifiers are mostly derived from the majority class. In this paper, we describe an ensemble learning method combined with active example selection to resolve the imbalanced data problem. Our method consists of three key components: 1) an active example selection algorithm to choose informative examples for training the classifier, 2) an ensemble learning method to combine variations of classifiers derived by active example selection, and 3) an incremental learning scheme to speed up the iterative training procedure for active example selection. We evaluate the method on six real-world imbalanced data sets in biomedical domains, showing that the proposed method outperforms both the random under sampling and the ensemble with under sampling methods. Compared to other approaches to solving the imbalanced data problem, our method excels by 0.03-0.15 points in AUC measure.

Index Terms—Bioinformatics, classification, interactive data exploration and discovery, mining methods and algorithms.

1 Introduction

Machine learning techniques have been used in many real-world domains such as the Internet, scientific and business studies, and industry applications. Biomedical data are one of the popular domains of applications. When we train a classifier from data, training data sometimes have imbalanced class distribution [1], [2], [3], [4]. The imbalanced data problem occurs when class examples are inherently rare or hard to collect. For example, biomedical data derived from rare disease and abnormal prognosis are difficult to obtain, and some biomedical data are often obtained via expensive experiments. Since most machine learning algorithms train a classifier based on the assumption that the number of training examples of classes is almost equal, when we apply machine learning algorithms to imbalanced data, the trained classifier is mostly derived from the majority class. Additionally, we may miss or ignore essential patterns (i.e., information) from the minority class; this results in very poor prediction performance of the minority class because training the minority class is not done. In many cases, the user is more interested in minority classes. Thus, addressing and solving imbalanced data problem is very critical for improving classification performance.

- *S. Oh is with the WISE Lab., Division of Information and Computer Engineering, Ajou University, Suwon, Kyeonggi 443-749, Korea. E-mail: syoh@ajou.ac.kr.*
- *M.S. Lee and B.-T. Zhang are with the Center for Biointelligence Technology (CBIT) and School of Computer Science and Engineering, Seoul National University, Seoul 151-742, Korea. E-mail: {mslee, btzhang}@bi.snu.ac.kr.*

Manuscript received 29 Apr. 2010; revised 12 July 2010; accepted 13 July 2010; published online 22 Sept. 2010.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBBSI-2010-04-0108.
Digital Object Identifier no. 10.1109/TCBB.2010.96.

Since reliability and performance of an output model depend on the quality of training data, the sufficient amount of informative examples is essential to learn a good classifier. Most training data include some redundant examples or less useful examples. This may lead to performance degradation or long training time. Examples in the imbalanced data may exist redundantly, or some of them may be less useful.

In this paper, we propose a novel scheme to solve the imbalanced data problem, ensemble learning based on active example selection (EAES). The main part of our proposed scheme is the Active Example Selection (AES) method. AES builds a classifier by starting from a small balanced subset of training data and training a classifier iteratively through adding useful examples into the current training set. Even though AES performs well for improving imbalanced classification performances, AES also has weaknesses. Its iteration of model training and example selection steps requires lots of computation cycles, and thus the cost of using this method is high. Also, its output classifiers can vary depending on the initial training examples.

To address AES's high computational cost from the iterative model training and avoid possible a biased decision of AES, we improve the AES method with an incremental learning algorithm and an ensemble learning method. For the proposed EAES, we use the incremental naïve Bayes algorithm as a base classifier of AES instead of the iterative batch one. As a result, we make the training time of AES shorter than the time of the iterative batch learning algorithm. Additionally, we build an ensemble model by connecting various classifiers from different initial training examples to reduce the variance of classification models derived from AES and to get a robust output classifier. By integrating the different predictions from individual classifiers, the ensemble model can increase classification performance along with avoiding biased decisions.

This paper is organized as follows: in Section 2, we present related works. We present our AES technique that is the basis of the proposed EAES in Section 3. The overall EAES scheme is presented in Section 4. In Section 5, we show our empirical experiments and evaluate the results. We conclude in Section 6.

2 Related Works

Recent research on the imbalanced data problem has been focused on several major groups of techniques. The popular method to solve imbalanced data problem balances the number of training examples among classes via the resampling examples. To balance the number of training examples among classes, the random under sampling (RUS) randomly discards examples of a majority class, while the random over sampling (ROS) duplicates examples in a minority class. We can combine these two techniques to apply oversampling for the minority class and under sampling for the majority class, respectively. These random resampling techniques are easy to apply and improve the performance of classifiers by compensating for the imbalanced class distribution. However, they also produce unwanted effects such as overfitting or information loss by duplicating or deleting examples from the training sets using the techniques. To overcome these imbalanced data problems of random resampling, several new techniques have been introduced using the intelligent approaches (e.g., creating new examples for the minority class that is inferred from existing examples [5] and removing noise or duplicated examples from the majority class [6]). However, according to comparative studies of various resampling techniques, rather simple RUS or ROS generally performs better than new intelligent techniques mentioned above [7], [8].

There are several methods proposed to solve the imbalanced data problem using the ensemble of undersampled classifiers [9], [10]. These methods are usually based on the bagging, and we draw almost the same numbers of majority and minority examples for the sampled subset data. They fully utilize the minority and majority examples.

Among such methods, there is an active learning approach based on the support vector machine algorithm to solve imbalanced data problem. Typically, the class imbalance ratio of examples that are close to the decision boundary is lower than the imbalance ratio in the complete data set. Therefore, the active learning approach provides more balanced training examples because it selects examples that lie closest to the separating hyperplane using the support vector machine algorithm [11], [12]. However, the method is designed based on the characteristics of the support vector machine algorithm; thus, the method cannot be applied to other classification algorithms. In addition, the study is limited to binary problems that are simple enough to analyze, because the method selects examples that lie closest to the decision boundary.

Biomedical domain is our main focus of application in this paper. Here are some recent important studies about handling imbalanced biomedical data problem: one frequently used method is dividing the original data set into a balanced data set for training and an imbalanced data set for testing. We can avoid the imbalanced data problem by using a balanced data set for training. The method is used to diagnose myocardial perfusion using cardiac Single Proton Emission Computed Tomography (SPECT) images [13] and to predict polyadenylation signals in human sequences [14].

For imbalanced biomedical data, RUS techniques can also be easily applied. To discriminate deleterious nsSNPs (nonsynonymous single nucleotide polymorphisms) from neutral nsSNPs with an imbalanced training data set, prediction performances are improved by applying the RUS method combined with a decision tree algorithm [15]. Moreover, classifiers from the RUS method can be combined together into an ensemble machine (ERUS). An ensemble of undersampled classifiers is constructed for predicting the activity of drug molecules based on the structural characteristics of compounds [16] and for predicting glycosylation sites in genomic sequences [17].

3 Active Example Selection

To address the imbalanced data problem, AES iteratively collects useful training examples from the entire training data and excludes redundant or less useful examples. AES starts with randomly selected small number of examples that are balanced among classes and trains a classifier by incrementally adding useful examples [18]. By doing so as well as learning a classifier using informative examples, AES can efficiently solve the performance degradation problem that is caused by the imbalanced data. In this section, we describe how to derive a measure of usefulness to evaluate candidate examples in the AES procedure and overall learning procedure of AES.

3.1 A Measure of Usefulness

AES trains a classifier by adding useful examples iteratively. Thus, we first need to define the measure of usefulness of an example. The criterion for selecting useful examples for AES can be derived as follows:

First, let's assume that a base classification algorithm of AES is determined. Then, the training set of input-output pairs is given:

$$D_N = \{(x_1, y_1), \ldots, (x_N, y_N)\}. \tag{1}$$

To achieve good generalization performance, the objective functions of most machine learning algorithms are designed to directly minimize the additive error function on the data set D_N:

$$E(D_N|\boldsymbol{\theta}) = \sum_{p=1}^{N} E(y_p|x_p, \boldsymbol{\theta}). \tag{2}$$

However, our AES does not directly train a classifier using the entire training set of size N. Rather, AES starts the learning with a small subset of size $N_0 < N$ of given examples and increases the training set incrementally. In other words, we try to minimize the sequence of objective functions as follows:

$$E(D_{N_0}|\boldsymbol{\theta}_0), E(D_{N_1}|\boldsymbol{\theta}_1), \ldots, E(D_{N_s}|\boldsymbol{\theta}_s), \tag{3}$$

where N_i is the ith size of the training set satisfying the relation $N_0 < N_1 < \cdots < N_s = N$.

Second, let's assume that we have trained a classifier on the data set D_N. The classifier can be viewed as a function of the model parameter $\boldsymbol{\theta}$:

$$P_N(\boldsymbol{\theta}) = P(D_N|\boldsymbol{\theta}) = \prod_{p=1}^{N} P(y_p|x_p, \boldsymbol{\theta}), \tag{4}$$

where

$$P(y_p|x_p, \boldsymbol{\theta}) = \frac{\exp(-\beta E(y_p|x_p, \boldsymbol{\theta}))}{Z(\beta)}. \tag{5}$$

Here, β is a positive constant that determines the sensitivity of the probability to the error value, and Z(β) is a normalizing constant that is given by

$$Z(\beta) = \int \exp(-\beta E(y_p|x_p, \boldsymbol{\theta}))dy. \tag{6}$$

To improve the performance of the current classifier, AES incrementally expends the training data set by maximizing information gain of the classifier. The usefulness of a new example (x_{i+1}, y_{i+1}) can be determined by measuring the information gain of the classifier when we add the new example to the training data D_i. Let $P_i(\boldsymbol{\theta})$ and $P_{i+1}(\boldsymbol{\theta})$ be the probability distributions of the parameters before and after receiving the example, respectively. According to the information theory, the difference between $P_i(\boldsymbol{\theta})$ and $P_{i+1}(\boldsymbol{\theta})$ is given by

$$I(P_{i+1}, P_i) = \int P_{i+1}(\boldsymbol{\theta}) \ln \frac{P_{i+1}(\boldsymbol{\theta})}{P_i(\boldsymbol{\theta})} d\boldsymbol{\theta}. \tag{7}$$

The greater value of $I(P_{i+1}, P_i)$ means less resemblance between the two distributions and more information gain about $\boldsymbol{\theta}$. Thus, for the given fixed distribution $P_i(\boldsymbol{\theta})$, the maximum information gain is achieved by maximizing the difference between $P_{i+1}(\boldsymbol{\theta})$ and $P_i(\boldsymbol{\theta})$.

Considering the relation between (4) and (5), we can maximize this $I(P_{i+1}, P_i)$ by selecting the example with which we can have the greatest $E(D_{i+1}|\boldsymbol{\theta})$ to the current parameters $\boldsymbol{\theta}$ when we add it to D_i. Hence, the example that maximizes

$$\Delta E_{i+1} = E(D_{i+1}|\boldsymbol{\theta}) - E(D_i|\boldsymbol{\theta}) \tag{8}$$

is useful to improve the performance of the current classifier. Also, the training method should be able to reduce the error to the level of desired accuracy.

We can find the most useful example by inputting the rest of total training examples to the partially trained classifier, computing their errors $E(y_p|x_p, \boldsymbol{\theta})$, and selecting the kth example satisfying

$$E(y_k|x_k, \boldsymbol{\theta}) = \max_p \{E(y_p|x_p, \boldsymbol{\theta})\}. \tag{9}$$

As an error function $E(y_p|x_p, \boldsymbol{\theta})$, we can use the sum of squared errors between the desired output y_p and the actual output $f(x_p; \boldsymbol{\theta})$ of the trained classifier. Therefore, we can define the usefulness of an example (x_p, y_p) as

$$e_\theta(x_p) = \frac{E(y_p|x_p, \boldsymbol{\theta})}{\dim(y_p)} = \frac{1}{m}\sum_{i=1}^{m} (y_{pi} - f_i(x_p; \boldsymbol{\theta}))^2, \tag{10}$$

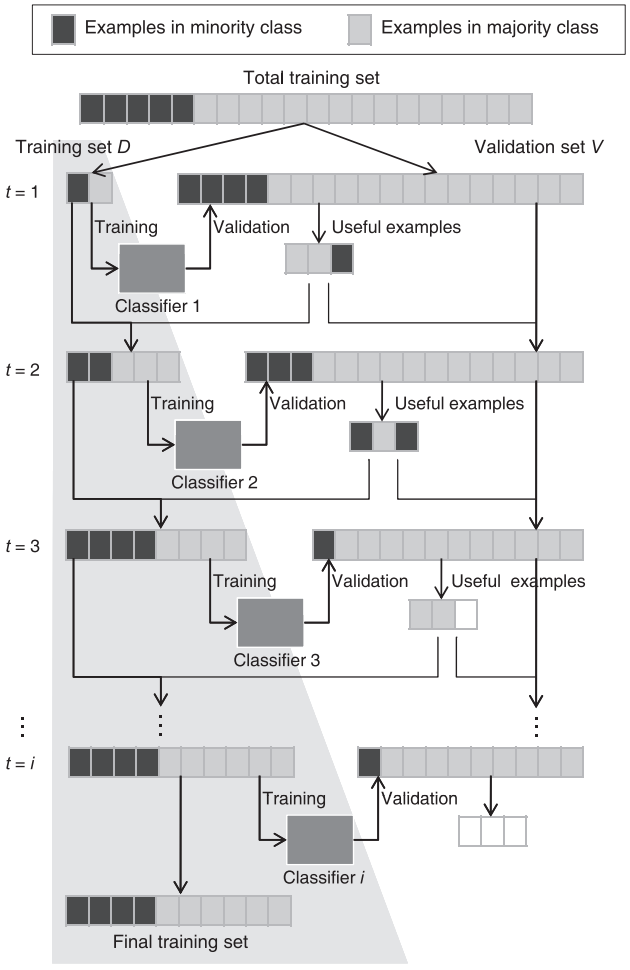

Fig. 1. An overview of the active example selection process.

where m indicates the number of target classes and f_i denotes the output prediction value of the ith target class. According to the definitions we made in (9) and (10), an example is the most useful if it causes the largest error on the current classifier.

3.2 Learning by Active Example Selection

AES is an active and incremental learning method to solve the imbalanced data problem. Fig. 1 depicts the overview of the AES process. Training set D is defined to be the set of data that are currently used to train a classifier. The rest of given examples is called the validation set V. D and V are initialized with a small set of randomly chosen seed examples and the rest of the given training examples, respectively.

The AES process is an iterative procedure that includes a training phase and an example selection phase. In the training phase, the model parameter $\boldsymbol{\theta}$ is updated using only the examples in the training set D. In the example selection phase, the examples in the current validation set V are tested by computing the usefulness $e_{\boldsymbol{\theta}}(x_p)$ of the examples (10) with respect to the current model $\boldsymbol{\theta}$. If $|V| > \lambda$, then λ numbers of the most useful examples (x_q, y_q) are selected from the validation set V and are moved to the training set D. Otherwise, all examples in V are selected into D. Using the updated training set D and validation set V, the next cycle of training and selection is done. AES will

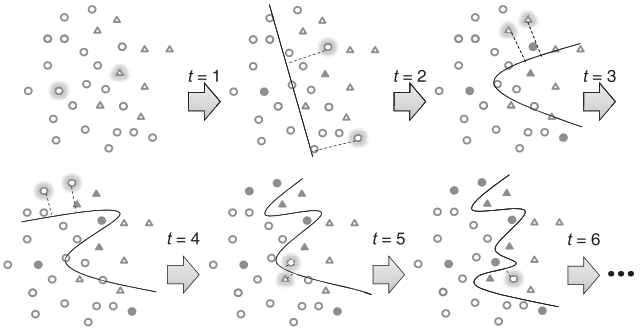

Fig. 2. An illustrative example of incremental growth of the training set D and changes of a trained model (depicted as a curve between two groups) in the binary and linearly separable classification problem.

terminate training when the specified performance level is achieved or the validation set V is empty. Note that the classifier has been generalized correctly to the validation set V if the algorithm halts with nonempty V.

Fig. 2 illustrates an example of AES in the binary and linearly separable classification problem. We set the parameters as the number of initial training example per class $= 1$ and the number of incremental chunk size $(\lambda) = 2$. In this figure, circles and triangles represent examples in the majority class and the minority class, respectively. Also, solid rectangles and empty rectangles represent examples in the current training set D and examples in the validation set V, respectively. Among the validation set V, shaded rectangles represent selected useful examples. In Fig. 2, we set an initial training example from each class (a solid circle and solid triangle). Then, we build a classifier using two initially selected examples and iteratively selected two examples (two shaded circles) based on (10) from the validation set $V[t = 1]$. After $t = 1$ stage, we update the classifier with the recently selected examples, validate the current classifier with the validation set V, and select the next two examples (two shaded triangles) $[t = 2]$. We iterate this procedure until the terminate criteria is satisfied.

4 Ensemble Learning Based on Active Example Selection (EAES)

In this section, we describe EAES and the incremental naïve Bayes classifier that is a base learner of EAES in detail. We combine the ensemble learning method with AES to avoid biased decisions to address possible variations of the resulting model that is caused by the composition of selected examples.

4.1 The Proposed Method: EAES

AES resolves the imbalanced data problem nicely by iteratively selecting useful examples and updating a current classifier. An output classifier is resulted from the initial training examples that are just a small part of the entire training data. However, since used examples cover a part of sample space, slight changes to the initial training data may easily lead to changes in the output model. Therefore, we improve the classification performance of AES by using the ensemble learning method (EAES). Ensemble learning methods combine multiple models and use them as a committee for decision making. While doing so, the ensemble learning method increases memory and computational cost. Nevertheless, the ensemble learning method mostly increases prediction performance over a single model, because it reduces the variance of prediction errors and avoids biased decisions [19].

Model generation
- For each of K iterations:
  - Perform AES.
    - Initialize a component classifier using a predefined classification algorithm.
    - Randomly select initial training examples per class.
    - Until meet termination conditions:
      - Update or re-train the classifier using selected examples.
      - Validate current classifier with the rest of the give training examples.
      - Select useful examples among them by (10).

Classification
- For each of the K models:
  - Assign weight to each model (12) using its training error.
  - Predict a class label of an example using weighted voting.
- Return the class label with highest voting score using (11).

Fig. 3. Pseudocoded EAES algorithm.

EAES builds an ensemble of component classifiers learned with different composition of training data that is derived from AES. Since each component classifier is trained with different compositions of initial training examples, diverse subsets of the original training data are used for training each component classifier. Each of the component classifiers covers a different part of sample space. Thus, the resulting ensemble model can improve generalized prediction performance.

EAES works similarly to bagging that is a popular ensemble learning method that uses different subsets of training data with a single learning method. However, there are two differences between EAES and bagging. EAES uses training subsets that are derived from AES with different compositions of the initial training examples. On the other hand, bagging uses randomly selected examples with replacement until the number of the selected examples is equal to the number of the original training set. Also, since the selected examples from AES are the strict subset of the original training data, they do not contain duplicates of the original training set. On the other hand, bagging generates a training set that contains some redundant examples.

For making the final decision, we use the weighted voting policy in EAES. In detail, EAES computes the final decision using the weighted sum of classification results with prediction probability distribution and weights derived from classifiers' training performance.

The EAES algorithm can be pseudocoded as shown in Fig. 3. As depicted, EAES trains K component classifiers based on AES using randomly selected initial training data. In this paper, we use the incremental naïve Bayes classifier as a base learner of AES (we describe this in the following section).

After that, EAES binds K component classifiers from K different training subsets to make a final classification result. EAES calculates the prediction result using weighted voting.

Let $\mathbf{x}$ be a query example, and $\boldsymbol{\theta}_i (i = 1, \ldots, K)$ be a parameter vector of the ith component classifier from AES. To get the target class of the query example $\mathbf{x}$, we can calculate it as follows:

$$f(\mathbf{x}) = \arg\max_{c \in C} \sum_{i=1}^{K} \alpha_i P_i(c|\mathbf{x}, \boldsymbol{\theta}_i), \quad (11)$$

where α_i is calculated based on the training error rate ε_i of the ith component classifier of the form

$$\alpha_i = \frac{\exp(-\varepsilon_i)}{\sum_{i=1}^{K} \exp(-\varepsilon_i)}. \quad (12)$$

4.2 Base Learner: Incremental Naïve Bayes Classifier

The AES can be applied as a wrapper learner of classification algorithms that produces a predicted class with a confidence value. The AES performs well with classification algorithms which have a small number of parameters and takes a short training time. Since AES expands the training data by the iterative procedures of training and validating a classifier, classification algorithms that can be implemented using an incremental learning procedure are very suitable as a base learner for AES. We define that a learning task is incremental if the training examples become available over time and are usually used one at the time. We also define that a learning algorithm is incremental if, for any given training sample $x_1, \ldots, x_n$, it produces a sequence of hypotheses $h_0, h_1, \ldots, h_n$, such that h_{i+1} depends only on h_i and the current example x_i [20]. In this paper, we use the incremental naïve Bayes classifier as a base learner of AES.

The naïve Bayes classifier is a simple probabilistic classifier based on Bayes' theorem. In particular, the naïve Bayes classifier assumes that the predictive attributes are conditionally independent of the given class and it hypothesizes that no hidden or latent attributes influence the prediction process [21]. These assumptions make the classification algorithm efficient. Let c be the random variable denoting the class of an example and let $\mathbf{x}$ be an observed example. Also, let x_i represent the ith attribute of $\mathbf{x}$. The naïve Bayes classifier selects the class label c^* with the maximum probability that is calculated according to the following equation:

$$c^* = \arg\max_{c \in C} P(c|\mathbf{x}). \quad (13)$$

Using Bayes' theorem, we can rewrite this equation as

$$c^* = \arg\max_{c \in C} \frac{P(c)P(\mathbf{x}|c)}{P(\mathbf{x})}. \quad (14)$$

Since the denominator $P(\mathbf{x})$ is the same for all classes and does not affect the relative values of their probabilities, we can ignore it. Then, we can have

$$c^* = \arg\max_{c \in C} P(c)P(\mathbf{x}|c). \quad (15)$$

$P(\mathbf{x}|c)$ can be decomposed into the product of $P(x_1|c), \ldots, P(x_a|c)$, since we assume the conditional independency among attributes x_1 through x_a given the class. Therefore, we can get the final formula for the Naïve Bayes classifier as

$$c^* = \arg\max_{c \in C} P(c) \prod_j P(x_j|c). \quad (16)$$

The final MAP (Maximum a Posteriori) decision rule (16) predicts the most probable class of a given example $\mathbf{x}$. All class priors $P(c)$ and attribute conditional probability distributions $P(x_j|c)$ can be approximated using the relative frequencies in training data. Each of the $P(c)$ can be estimated by counting the frequency of each target value c in the training data. To compute the conditional probability in the case of nominal attributes, we only need to maintain a counter for each attribute value and for each class. In the case of continuous attributes, we can assume a particular distribution for the values of attribute or discretize the attribute in a preprocessing phase.

Despite its naïve design and oversimplified assumptions, the naïve Bayes classifier shows good performances in many complex real-world problems. Moreover, the naïve Bayes classifier requires a small amount of training data for parameter estimation. Since independent attributes are assumed, only the variances of the attributes of each class need to be determined and not the entire covariance matrix. All the probabilities required to solve (16) can be computed from the training data with one step. As a result, it leads to low computational cost and relatively low memory consumption.

Another interesting aspect of the algorithm is its easy implementation in an incremental fashion because only counters are used. The naïve Bayes classifier builds a table for each attribute. The table reflects the distribution on the training data of the attribute values over the classes. The incremental naïve Bayes classifier is initialized with zero training examples. Then, it can learn incrementally using one example at a time by updating the tables. The trained incremental naïve Bayes classifier can be utilized by calculating the class membership probabilities for the given test example based on the tables.

AES works well with the incremental naïve Bayes classifier because it has a small number of parameters to be tuned and spends a short training time. In addition, incremental learning algorithms [20] are very suitable for being incorporated with iterative procedure of AES.

5 Evaluations

In this section, we present the empirical results that show the performance of AES and EAES with imbalanced biomedical data. We also analyze the experimental results here.

5.1 Data Sets and Experimental Setup

We perform empirical experiments using six real-world biomedical benchmark data sets: hepatitis clinical data (Hepatitis) [24], [25], voice data of Parkinson's disease (Parkinson) [26], diabetes clinical data (Diabetes) [27], image data of prognostic breast cancer (WPBC) [28], image data of cardiac disease (SPECT) [13] from UCI machine learning repository [22], and public leukemia microarray data (Leukemia) [23].

We focus on the binary classification problems in this paper. The overview of the used data sets for the experiments is given in Table 1. As the data preprocessing steps continue, a range of numeric attributes in the data set is discretized into nominal attributes for the naïve Bayes classifier.

To investigate the performances of the proposed AES and EAES, we conducted experiments to compare

1. the naïve Bayes classifier algorithm (NB),
2. NB with RUS,

TABLE 1
Overview of Data Sets

Dataset	# of examples	# of attributes	Class distribution	Imb. ratio
Hepatitis	155	19	terminal: 32 survival:123	1:3.84
Parkinson	194	22	healthy: 47 disease: 147	1:3.13
Diabetes	768	8	diabetes: 268 healthy: 500	1:1.87
WPBC	198	33	recur: 47 non-recur: 151	1:3.12
SPECT	267	43	normal: 55 abnormal: 212	1:3.85
Leukemia	72	7129	AML: 25 ALL: 47	1:1.88

3. NB with AES,
4. ensemble of RUS (ERUS), and
5. ensemble of AES (EAES).

Since AES produces a subset of the training set, we compare the performance of AES with that of RUS that produces a randomly undersampled subset of the training set. We choose RUS because it generally shows better performance than new intelligent approaches [7]. EAES is an ensemble method based on AES; thus, we compare the performance of EAES with that of ERUS that is an ensemble method based on RUS. ERUS is chosen because it was used to solve many biomedical imbalanced data problems [16], [17]. RUS and ERUS are incorporated with the naïve Bayes classifier. To shorten training time, AES and EAES are incorporated with the incremental version of the naïve Bayes classifier.

The parameters of the AES procedures are set for all data as follows: the number of initial training example per class is 1 and the incremental example size is 2. In the ensemble learning (i.e., ERUS and EAES), the number of component classifiers is set to 15.

To evaluate the performance of classification methods, AUC (Area under the ROC Curve), overall accuracy, and true positive rate are used. When a data set is highly skewed, the overall accuracy tends to be overwhelmed by the prediction power of the majority class. In this case, the comparison of the overall accuracy is very much misleading. Because of this, we used the AUC that gives balanced evaluation by measuring both positive and negative classes with equal weights. For the imbalanced data problem, the AUC has been widely used as a performance evaluation measure. In addition, we use the true positive rate (TPR) as an evaluation measure which represents the classification performance per class. The true positive rates are computed by the ratio of correctly predicted examples of a class among all available examples of the class during the test.

To estimate the general performances of AES and EAES, 10-fold cross validations were executed for each combination of six data sets and five learning strategies. We averaged the performance of the total runs for each combination with standard deviation. The results are shown in Tables 2, 3, 4, 5, 6, and 7.

TABLE 2
The Data Efficiency of AES Training

Dataset	Original training examples (imbalance ratio)	Used examples for training (imbalance ratio)	Data efficiency
Hepatitis	Terminal: 28.8 Survive: 110.7 (1:3.84)	Terminal: 10.0 Survival: 14.7 (1:1.47)	17.7%
Parkinson	Healthy: 42.3 Disease: 132.3 (1:3.12)	Healthy: 11.9 Disease: 44.8 (1:3.76)	32.5%
Diabetes	Diabetes: 241.2 Healthy: 450.0 (1:1.87)	Diabetes: 116.7 Healthy: 117.1 (1:1.0)	33.8%
WPBC	Recur: 42.3 Non-recur: 135.9 (1: 3.21)	Recur: 25.9 Non-recur: 51.3 (1:1.98)	43.9%
SPECT	Normal: 49.5 Abnormal: 190.8 (1:3.85)	Normal: 15.9 Abnormal: 62.5 (1:3.93)	32.6%
Leukemia	AML: 22.5 ALL: 42.3 (1:1.88)	AML: 5.4 AML: 5.4 (1:1.0)	16.7%

TABLE 3
The Correlation Analysis between False Negative Rate and the Number of Added Examples

Dataset	Pearson's correlation coefficient	
	Minority class	Majority class
Hepatitis	Terminal 0.263±0.162	Survive 0.330±0.110
Parkinson	Healthy -0.085±0.246	Parkinsons 0.436±0.105
Diabetes	Diabetes 0.287±0.056	Healthy 0.336±0.041
WPBC	Recur 0.115±0.108	Non-recur 0.258±0.069
SPECT	Normal -0.069±0.150	Abnormal 0.353±0.050
Leukemia	AML 0.502±0.199	ALL 0.392±0.116

5.2 Evaluation of the Results

In this section, we examine how AES works in imbalanced biomedical data sets and evaluate the performance of AES and EAES by comparing them with other methods.

TABLE 4
The Comparison of AUC

Dataset	NB	RUS	AES	ERUS	EAES
Hepatitis	0.86±0.08	0.88±0.07	0.92±0.03	0.89±0.07	**0.94**±0.03
Parkinson	0.85±0.12	0.86±0.11	0.91±0.07	0.86±0.11	**0.92**±0.09
Diabetes	0.81±0.04	0.82±0.04	**0.84**±0.04	0.82±0.03	**0.84**±0.03
WPBC	0.69±0.13	0.66±0.15	0.79±0.07	0.67±0.10	**0.84**±0.14
SPECT	0.86±0.08	0.85±0.07	**0.91**±0.05	0.86±0.08	0.90±0.04
Leukemia	**1.00**±0	**1.00**±0	**1.00**±0	**1.00**±0	**1.00**±0

TABLE 5
The Comparison of Accuracy (Percent)

Dataset	NB	RUS	AES	ERUS	EAES
Hepatitis	83.7±9.2	81.8±8.7	**93.5**±8.1	81.1±9.2	92.8±9.0
Parkinson	68.7±10.0	68.7±10.4	76.8±9.2	69.2±10.1	**81.4**±9.1
Diabetes	75.1±4.0	74.5±2.3	77.9±4.3	74.3±3.1	**78.4**±4.3
WPBC	67.2±7.9	61.7±11.9	73.7±8.7	61.6±7.6	**81.8**±7.8
SPECT	68.9±5.9	66.0±6.4	77.2±5.6	66.3±4.7	**79.8**±6.6
Leukemia	**100.0**±0	**100.0**±0	**100.0**±0	98.8±4.0	**100.0**±0

TABLE 6
The Comparison of True Positive Rate per Class (Percent)

Dataset	Class	NB	RUS	AES	ERUS	EAES
Hepatitis	Min	67.5±27.3	77.5±27.2	**87.5**±16.3	77.5±22.2	**87.5**±16.3
	Maj	87.7±10.6	82.9±9.6	**95.1**±8.8	82.1±10.2	94.3±11.0
Parkinson	Min	86.5±19.2	89.0±19.1	**95.5**±9.6	91.0±19.1	93.5±10.6
	Maj	62.5±11.7	61.9±12.5	70.6±11.5	61.9±12.9	**77.4**±9.9
Diabetes	Min	60.1±9.8	68.6±4.7	68.7±7.1	68.3±5.5	**70.2**±8.1
	Maj	**83.2**±4.2	77.6±2.5	82.8±3.9	77.6±4.2	82.8±3.7
WPBC	Min	47.0±17.2	53.0±17.2	57.5±18.7	56.0±16.3	**62.0**±20.3
	Maj	73.4±9.6	64.1±16.5	78.8±8.1	63.5±7.5	**88.0**±8.2
SPECT	Min	87.7±11.7	93.0±9.1	96.7±7.0	93.0±9.1	**100**±0
	Maj	64.2±8.3	59.0±8.6	72.2±7.0	59.5±5.7	**74.6**±8.1
Leukemia	Min	**100.0**±0	**100.0**±0	**100.0**±0	**100.0**±0	**100.0**±0
	Maj	**100.0**±0	**100.0**±0	**100.0**±0	98.0±6.3	**100.0**±0

5.2.1 Characteristics of AES on Imbalanced Data

To characterize the learning procedure of AES, we investigate learning curves of AES and patterns in selected examples during training a classifier using AES.

Fig. 4 demonstrates examples of training, validation, and test curves of AES. The plots show AUC performances of each iteration step which are drawn with total training set $D+V$, validation set V, and independent test set. These curves are from one of 10 runs for each data set. In the early stage of incremental learning, the first approximation may be not satisfactory. However, the next set of useful examples can be selected using this knowledge. The selected examples may cause oscillations in learning curves. However, the learning curves are improved steadily as iterations go on. When the AES learning is terminated, the validation AUC converges into 1. Even though the learned classifier based on AES seems to be overfitted to the validation set, as we can see in Fig. 4, the classifiers are not overfitted to the total training set.

The iterative procedure of AES learning can be terminated when the validation data are exhausted or there is no

TABLE 7
The Effects of Incremental Learning Strategy Execution Time (Seconds)

Dataset	# of examples	# of attributes	AES batch	AES inc.	EAES batch	EAES Inc.	AES Efficiency	EAES Efficiency
Hepatitis	158	19	0.2	0.1	3.6	2.2	34.4%	38.7%
Parkinson	194	22	0.8	0.7	15.4	13.4	11.1%	13.1%
Diabetes	768	8	6.5	4.3	131.7	86.8	33.6%	34.1%
WPBC	198	33	1.6	1.1	33.9	22.4	33.2%	33.8%
SPECT	267	43	2.8	5.3	57.5	46.4	17.3%	19.3%
Leukemia	72	7129	19.8	1.3	427.8	51.3	**93.2%**	**88.0%**

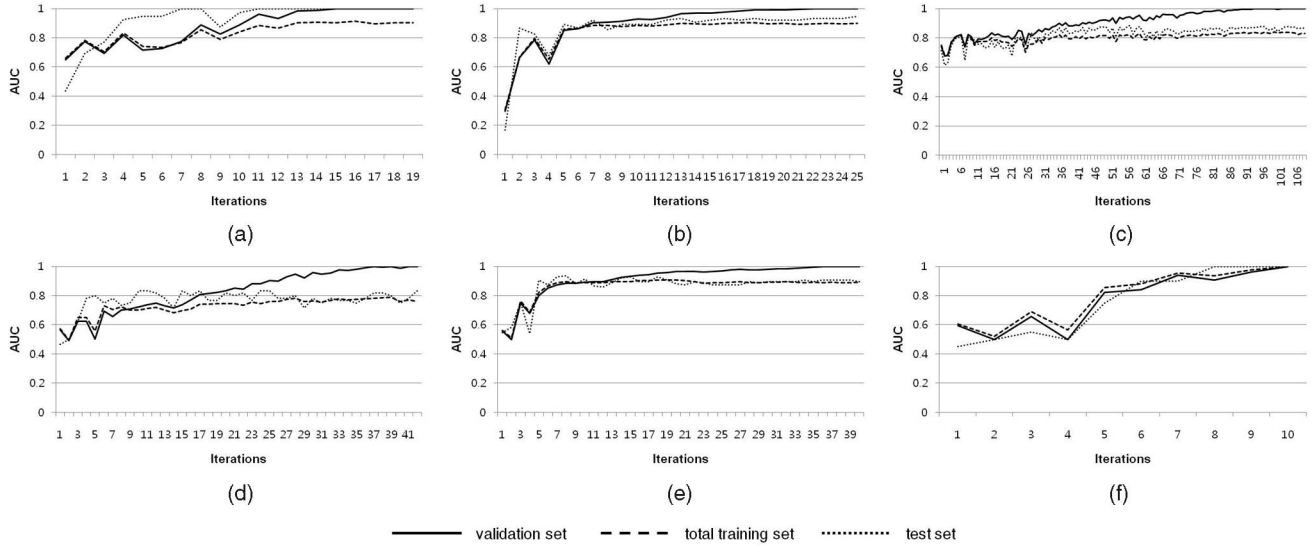

Fig. 4. The learning curves of AES with six real imbalanced biomedical data sets. Solid lines in graphs represent the validation performance with validation sets (V), dashed lines represent the training performance with total training sets $(D+V)$, and dotted lines represent the test performance with independent test sets.

misclassified example in the validation data. However, all classifiers derived from 10-fold cross-validation with six data sets are terminated in case with the absence of misclassified examples in the validation set. Table 2 shows data efficiency of AES training. Table 2 indicates the average number of original training examples except test examples as well as the average number of the used examples for the last training iteration of AES in 10-fold cross validation. The data efficiency percentages of AES (i.e., the percentage of the number of used examples for training to the number of original training examples) range from 16.7 percent to 43.9 percent. By the active example selection process, AES effectively excludes the less important or redundant training examples. Table 2 also depicts changes of the imbalance ratios between the original training data sets and used examples with the AES method. Most methods to solve the imbalanced data problem are balancing the training data distribution. However, the AES method does not consider the balance of class distribution but only consider the prediction performances. Hence, the imbalance ratios of used examples in Table 2 are not always decreased.

The false negative rate indicates that the proportion of positive examples was erroneously classified as negative. If a class has a high false negative rate, then it indicates that the training of the class has not been completed. To characterize the AES operation for handling the imbalanced data problem, we conduct a correlation analysis between the number of added examples and false negative rate on an iterative AES learning process as shown in Table 3. By investigating the correlation between the number of added examples and false negative rate for each class, we can verify whether the weakness of the current classifier is reflected when the active example selection incrementally expands training examples or not. Note that the range of a number of added examples is from 0 to 2 and the range of false negative rate is from 0 to 100. Even though the ranges of the two attributes are very different, the correlation analysis results using Pearson's correlation coefficients show relatively high positive correlations between them in most cases. The positive correlations indicate that the AES procedure identifies the weakness of the current classifier and adds useful examples of a high error score to the current classifier regardless of the original class distribution.

When we run the procedure of AES repeatedly, the classifier is improved efficiently using an even small subset of the training data set. As a consequence, AES resolves the imbalanced data problem by selecting useful examples based on information gain of the current classifier.

5.2.2 Discussions on Performance Comparisons

To show the effectiveness of the proposed AES and EAES methods, we compare the prediction performances of five methods (i.e., methods we describe in Section 5.1.2) with six imbalanced biomedical data sets (i.e., data sets we describe in Section 5.1.1).

From the experimental results, we find interesting issues to discuss. First, we argue that our EAES and AES mitigate the imbalanced data problem and achieve superior classification performances compared to RUS and ERUS. The improvement in AUC by 0.04-0.15 implies that EAES effectively deals with the imbalanced data problem (Table 4). The AUC of EAES is higher than that of AES, and it indicates that the proposed EAES reduces the possibility of distorting the data distribution. The distortion is caused by training a model using a subset of total data. Also, the improvement in accuracy by 3.3-14.6 percent implies that EAES upgrades the general performance of the output classifier by employing several classifiers as a decision committee (Table 4).

Second, our empirical study shows that a real imbalanced data problem is not an imbalanced class distribution but an imbalanced amount of information. In terms of true positive rate per class, the majority class does not always achieve higher prediction performance than the minority class as shown in Table 6. In the Parkinson and SPECT data sets, the true positive rate of the majority class is lower than that of the minority class. When the true positive rate of a majority class is not good, AES selects more examples of the class regardless of balancing the training data distribution (Table 2). By adding useful examples to the current classifier, AES effectively strengthens the current existing classifier and improves the prediction performance of each class. In addition, we can achieve better prediction performance in almost every case by combining ensemble learning with AES.

Third, by adopting the incremental learning algorithm, AES can simply update the current model with the selected examples without training all the examples repeatedly. As a result, we make the training time of AES shorter than the time of the iterative batch learning algorithm (Table 7). Since EAES includes several component classifiers that are trained based on the iterative AES procedure, the overall computational cost of EAES is strictly reduced by using the incremental learning algorithm. When a data set has a large number of attributes (e.g., microarray data—leukemia data set in Table 7), the incremental version of the naïve Bayes classifier strictly shortens the execution time.

Finally, our proposed methods outperform RUS and ERUS. They show slightly improved AUC by at most 0.03 (Table 4) and no improvement in the classification accuracy (Table 5). In terms of true positive rates of RUS and ERUS, the classification performance on the majority class is rather decreased by -5.2- -9.9 percent (Table 6). We presume that performance degradation on the majority class may be caused by the information loss by randomly discarding the majority class examples.

As we listed above, we found several reasons to verify that our proposed method outperforms the current imbalanced data problem solving methods by analyzing the results of our experiments. Even though experimental data sets are not covering every imbalanced data problem that exists currently, we believe that the current experiments are enough to show the effectiveness of our prosed methods.

6 Conclusion

In this paper, we described our AES and ensemble learning method based on AES (EAES) to solve the imbalanced data problem. Since some classes are not trained well when data are imbalanced, the imbalanced

data cause serious performance degradation for the classification. Examples in the imbalanced data may redundantly exist or some of them are less useful. Our AES solves the imbalanced data problem by iteratively collecting the useful training examples from the entire training data as well as by excluding redundant or less useful examples. By doing so and by learning a classifier using informative examples, AES can effectively mitigate the performance degradation caused due to the imbalanced data problem. To avoid biased decisions that may come from the composition of selected training examples, we introduce the ensemble learning method to AES (EAES). For speeding up the iterative AES procedure, we also adopt an incremental version of the naïve Bayes classifier algorithm.

To verify the effectiveness of our AES and EAES, we experiment with six real-world biomedical data sets, and the empirical results show that EAES and AES perform better than RUS and ERUS that are currently the most popular imbalanced data solving methods, and they strictly improve prediction performance.

While we analyzed the results, we found that the most important factor for improving classification performance is not balancing the number of examples between classes, but balancing amount of information which can be used in training. In other words, it is important to select and utilize informative examples when you have an imbalanced data problem. In the results, we can see that AES and EAES improve the prediction performance, while the imbalance ratios of used examples are not always decreased.

We expect that our EAES and AES can be applied to other real-world data mining applications, where we suffer from the imbalanced data problem. Also, our EAES can be used to identify discriminative or representative examples of some classes by investigating selected training examples that commonly appear among various AES runs.

ACKNOWLEDGMENTS

This work was jointly supported by the Ajou University Research Fellowship of 2009 (S-2009-G0001-00054), the MKE under the ITRC support program of the Korean Government supervised by the NIPA (NIPA-2010-C1090-1021-0011), the IT R&D Program of MKE/KEIT (KI002138, MARS), the NRF Grant of MEST (314-2008-1-D00377, Xtran), and the BK21-IT program of MEST. This paper has been significantly revised from an earlier version presented at the IEEE International Conference on Bioinformatics and Biomedicine 2009 (BIBM 2009) in November 2009. Sangyoon Oh and Min Su Lee contributed equally to this work. Byoung-Tak Zhang was the corresponding author for this paper.

REFERENCES

[1] N.V. Chawla, N. Japkowicz, and A. Kolcz, "Editorial: Special Issue on Learning from Imbalanced Data Sets," *ACM SIGKDD Explorations Newsletter,* vol. 6, no. 1, pp. 1-6, June 2004.

[2] G.M. Weiss, "Mining with Rarity: A Unifying Framework," *ACM SIGKDD Explorations Newsletter,* vol. 6, no. 1, pp. 7-19, June 2004.

[3] N. Japkowicz and R. Holte, "Workshop Report: AAAI-2000 Workshop Learning from Imbalanced Data Sets," *AI Magazine,* vol. 22, no. 1, pp. 127-136, 2001.

[4] ICML 2003 Workshop Learning from Imbalanced Data Sets (II), http://www.site.uottawa.ca/~nat/Workshop2003/workshop2003.html, 2010.

[5] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique," *J. Artificial Intelligence Research,* vol. 16, pp. 321-357, 2002.

[6] M. Kubat and S. Matwin, "Addressing the Curse of Imbalanced Training Sets: One Sided Selection," *Proc. 14th Int'l Conf. Machine Learning,* pp. 179-186, 1997.

[7] J.V. Hulse, T.M. Khoshgoftaar, and A. Napolitano, "Experimental Perspectives on Learning from Imbalanced Data," *Proc. 24th Int'l Conf. Machine Learning,* pp. 935-942, 2007.

[8] G.E.A.P.A. Batista, R.C. Prati, and M.C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," *ACM SIGKDD Explorations Newsletter,* vol. 6, no. 1, pp. 20-29, June 2004.

[9] S. Hido and H. Kashima, "Roughly Balanced Bagging for Imbalanced Data," *Proc. SIAM Int'l Conf. Data Mining,* pp. 143-152, 2008.

[10] P. Kang and S. Cho, "EUS SVMs: Ensemble of Under-Sampled SVMs for Data Imbalance Problems," *Lecture Notes in Computer Science,* pp. 837-846, Springer, Oct. 2006.

[11] S. Ertekin, J. Huang, L. Bottou, and C.L. Giles, "Learning on the Border: Active Learning in Imbalanced Data Classification," *Proc. ACM Conf. Information and Knowledge Management (CIKM '07),* pp. 127-136, Nov. 2007.

[12] S. Ertekin, J. Huang, and C.L. Giles, "Active Learning for Class Imbalance Problem," *Proc. ACM SIGIR '07),* pp. 823-824, July 2007.

[13] L.A. Kurgan, K.J. Cios, R. Tadeusiewicz, M. Ogiela, and L. Goodenday, "Knowledge Discovery Approach to Automated Cardiac SPECT Diagnosis," *Artificial Intelligence in Medicine,* vol. 23, no. 2, pp. 149-169, Oct. 2001.

[14] H. Liu, H. Han, J. Li, and L. Wong, "An In-Silico Method for Prediction of Polyadenylation Signals in Human Sequences," *Proc. 14th Int'l Conf. Genome Informatics,* vol. 14, pp. 84-93, Dec. 2003.

[15] R.J. Dobson, P.B. Munroe, M.J. Caulfield, and M.A.S. Saqi, "Predicting Deleterious nsSNPs: An Analysis of Sequence and Structural Attributes," *BMC Bioinformatics,* vol. 7, pp. 217-225, 2006.

[16] G.-Z. Li, H.-H. Meng, W.-C. Lu, J.Y. Yang, and M.Q. Yang, "Asymmetric Bagging and Feature Selection for Activities Prediction of Drug Molecules," *BMC Bioinformatics,* vol. 9, suppl. 6, p. S7, Aug. 2007.

[17] C. Caragea, J. Sinapov, A. Silvescu, D. Dobbs, and V. Honavar, "Glycosylation Site Prediction Using Ensembles of Support Vector Machine Classifiers," *BMC Bioinformatics,* vol. 8, pp. 438-450, Nov. 2007.

[18] M.S. Lee, J.-K. Rhee, B.-H. Kim, and B.-T. Zhang, "AESNB: Active Example Selection with Naïve Bayes Classifier for Learning from Imbalanced Biomedical Data," *Proc. IEEE Int'l Conf. Bioinformatics and Bioeng.,* pp. 15-21, 2009.

[19] E. Bauer and R. Kohavi, "An Empirical Comparison of Voting Classification 37 Algorithms: Bagging, Boosting, and Variants," *Machine Learning,* vol. 36, nos. 1/2, pp. 105-139, 1999.

[20] C. Giraud-Carrier, "A Note on the Utility of Incremental Learning," *AI Comm.,* vol. 13, no. 4, pp. 215-223, Dec. 2000.

[21] W.L. Buntine, "Operations for Learning with Graphical Models," *J. Artificial Intelligence Research,* vol. 2, pp. 159-225, 1994.

[22] A. Asuncion and D.J. Newman UCI Machine Learning Repository, http://www.ics.uci.edu/~mlearn/MLRepository.html, 2007.

[23] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science,* vol. 286, pp. 531-537, 1999.

[24] P. Diaconis and B. Efron, "Computer-Intensive Methods in Statistics," *Scientific Am.,* vol. 248, pp. 116-128, 1983.

[25] G. Cestnik, I. Konenenko, and I. Bratko, "Assistant-86: A Knowledge Elicitation Tool for Sophisticated Users," *Progress in Machine Learning,* I. Bratko and N. Lavrac, eds., pp. 31-45, Sigma Press, 1987.

[26] M.A. Little, P.E. McSharry, S.J. Roberts, D.A.E. Costello, and I.M. Moroz, "Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection," *BioMedical Eng. OnLine,* vol. 6, no. 23, pp. 23-42, June 2007.

[27] J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, and R.S. Johannes, "Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus," *Proc. Symp. Computer Applications and Medical Care,* pp. 261-265, 1988.

[28] W.N. Street, O.L. Mangasarian, and W.H. Wolberg, "An Inductive Learning Approach to Prognostic Prediction," *Proc. Int'l Conf. Machine Learning*, pp. 522-530, 1995.

Sangyoon Oh received the BEng degree in mechanical design from Sungkyunkwan University, Korea, the MSc degree in computer science from Syracuse University, and the PhD degree in computer science from Indiana University, Bloomington, in 1995, 1999, and 2006, respectively. He is currently an assistant professor in the Department of Information and Computer Engineering at Ajou University, Korea. Prior to 2007, he worked as a research manager for SK Telecom, which is the biggest telecommunication company in Korea. He has coauthored three books and published more than 30 papers in international journals and conference proceedings. His research interests include semantic web, data mining, and large-scale software systems. He is a member of the IEEE.

Min Su Lee received the BS degree in mathematics, the MS degree in computer science and engineering, and the PhD degree in computer science and engineering from Ewha Womans University, Korea, in 2001, 2003, and 2007, respectively. She is currently a postdoctoral researcher at the Center for Biointelligence Technology (CBIT) and the School of Computer Science and Engineering at Seoul National University, Korea. Her research interests include active learning, data mining, bioinformatics, and machine learning.

Byoung-Tak Zhang received the BS and MS degrees in computer science and engineering from Seoul National University (SNU), Korea, in 1986 and 1988, respectively, and the PhD degree in computer science from the University of Bonn, Germany, in 1992. He is currently a professor with the School of Computer Science and Engineering and the Graduate Programs in Bioinformatics, Brain Science, and Cognitive Science, SNU, and directs the Biointelligence Laboratory and the Center for Biointelligence Technology. Prior to joining SNU, he was a research associate with the German National Research Center for Information Technology from 1992 to 1995. From August 2003 to August 2004, he was a visiting professor with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge. His research interests include probabilistic models of learning and evolution, biomolecular/DNA computing, and molecular learning/evolvable machines. He serves as an associate editor for the *IEEE Transactions on Evolutionary Computation*, *BioSystems*, and *Advances in Natural Computation*.