Integrating new data balancing technique with committee networks for imbalanced data: GRSOM approach

Danaipong Chetchotsak^{1,2} · Sirorat Pattanapairoj^{1,2} · Banchar Arnonkijpanich^{3,4}

Received: 11 October 2014 / Revised: 7 July 2015 / Accepted: 21 July 2015 / Published online: 31 July 2015
© Springer Science+Business Media Dordrecht 2015

Abstract To deal with imbalanced data in a classification problem, this paper proposes a data balancing technique to be used in conjunction with a committee network. The proposed data balancing technique is based on the concept of the growing ring self-organizing map (GRSOM) which is an unsupervised learning algorithm. GRSOM balances the data through growing new data on a well-defined ring structure, which is iteratively developed based on the winning node nearby the samples. Accordingly, the new balanced data still preserve the topology of the original data. The performance of our proposed method is evaluated using four real data sets from the UCI Machine Learning Repository and the classification performance is measured using the fivefold cross validation method. Classifiers with most common data balancing techniques, namely the Minority Over-Sampling Technique (SMOTE) and the Random under-sampling Technique (RT), are used as the baseline methods in this study. The results reveal that a committee of classifiers constructed using GRSOM performs at least as well as the baseline methods. The results also suggest that classifiers constructed using neural networks with the backpropagation algorithm are more robust than those using the support vector machine.

Keywords Imbalanced data · Committee networks · Growing ring self-organizing map · Classification

Introduction

In many real world classification problems, training data for constructing a classifier are not always balanced. Imbalanced data occurs when the number of training samples is considerably larger in one class than in the other class. As a result, most classifiers tend to learn the majority class well and ignore the minority class. This is because they were trained to minimize the overall error, resulting in focusing of learning on the class with more instances. Therefore, the classification performance on the majority class would be much better than that on the minority class. Many applications that address the class imbalance problem include detection of oil spills (Kubat et al. 1998), fraud detection (Fawcett and Provost 1997; Hilas and Mastorocostas 2008), credit cards (Chan et al. 1999), risk management (Daskalaki et al. 2006; Huang et al. 2006), tornado prediction (Adrianto et al. 2010), hard disk drive defect detection (Chetchotsak and Pattanapairoj 2010), and medical research (Cohen et al. 2006; Mazurowski et al. 2008; Ganji et al. 2010; Li et al. 2010). Moreover, there are several works that attempt to solve imbalanced data problems. Examples of such recent works are Yen and Lee (2009), Liu et al. (2011), Hwang et al. (2011), Ren (2012), Kang et al. (2012), and Yong (2012).

Since data can be represented by means of prototypes (Arnonkijpanich et al. 2011), the task of data balancing can be executed by prototype-based method such as self-organizing map (SOM). The SOM as proposed by Kohonen generalizes standard vector quantization by integrating the priorly fixed prototype topology. Accordingly, SOM can be

✉ Banchar Arnonkijpanich
abanchar@kku.ac.th

¹ Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen, Thailand

² System Modeling for Industry Research Group, Faculty of Engineering, Khon Kaen University, Khon Kaen, Thailand

³ Department of Mathematics, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand

⁴ The Centre of Excellence in Mathematics, CHE, Si Ayutthaya Rd., Bangkok 10400, Thailand

used to extract topological information embedded in the given data. The main goal of SOM is to transform patterns in input space which displays arbitrary dimensionality onto one or two-dimensional array of neurons, such that topological ordering and neighborhood preservation takes place (Arnonkijpanich et al. 2010). Because of this topological ordering, it assures that vectors closely located in input space are assigned to adjacent neurons on the topological map. In addition, SOM is categorized as an unsupervised version of a vector quantization technique with soft competition between the neurons. Each neuron has an associated reference vector indicating the position of the prototype. These prototypes are considered as centers of clusters in input space. The data topology is obtained by connecting each prototype to its neighbors. The topology is correct if data and prototypes are sufficiently dense. However, it is very difficult to predetermine the appropriate number of prototypes. Therefore, classical SOM should be improved by combining it with neural-growing networks such as growing ring self-organizing map (GRSOM). Such neighborhood based neural adaptation in GRSOM also reduces the influence of initialization. During training, the problem caused by insufficient prototypes can be solved with the process of increasing the number of prototype vectors. By using adaptive prototype learning, GRSOM based method provides the data topology fitted to the underlying data manifold which leads to reasonable data representation.

In this paper, we propose a data balancing technique to be used in conjunction with a committee network. The data balancing technique is used either to increase the number of minority samples or decrease the number of majority samples, so as to balance the whole data set. Here, the data balancing technique based on the concept of GRSOM is proposed. GRSOM has been successfully used to solve the traveling salesman problem (Bai et al. 2006a, b). GRSOM generates new data samples through the use of interpolation in a non-linear fashion, while preserving the topology of the original data. Similar to the most common data balancing technique, namely SMOTE (Chawla et al. 2002, 2003), GRSOM grows new samples based on the distance of the data. However, SMOTE generates synthetic samples through considering the k -nearest neighbors of each example in the minority class, while GRSOM grows new data on a ring structure which is developed based on visiting all the points of the data samples. The new grown data will be inserted near the best winning node of the ring structure. Thus, GRSOM can preserve the topology of the original data set. In this paper, we demonstrate how GRSOM is used to solve imbalanced data problems.

In addition, committee networks are used to improve classification performance in this paper. Several works have shown that using committee networks can improve

prediction performance in both regression and classification contexts (Parmanto et al. 1996; Chetchotsak and Twomey 2007; Nanthapodej and Chetchotsak 2009). The concept of the committee networks is that many heads are better than one. They typically consist of several neural networks serving as committee members that have different expertise and help each other to perform a classification task. The committee networks make a decision through employing a fusion rule that combines the decisions (outputs) of each committee member. Typically, the most common fusion rule is the majority voting scheme, where the decision of each committee member is equally important, and the final decision is based on the majority of the decisions.

Thus, we hypothesize that using GRSOM in conjunction with committee networks would help to improve classification performance for an imbalanced data problem. In this study, four data sets from the UCI Machine Learning Repository, Center for Machine Learning and Intelligent Systems are used to evaluate our proposed method. To analyze the effectiveness of GRSOM, we compare our method against the most common data balancing methods: SMOTE and RT. The rest of this paper is organized as follows. “[Related literature](#)” section presents related literature. “[Proposed framework](#)” section describes the proposed framework consisting of GRSOM-based data balancing and model construction phases. The experimental method is provided in “[Experimental method](#)” section. The experimental results and discussion are in “[Results and discussion](#)” section and finally “[Conclusion](#)” section concludes the paper.

Related literature

Several techniques are proposed to solve classification problems with imbalance of class distributions through data or algorithmic approaches or combinations of both (Fernandez et al. 2008). At the data level, the training data is balanced using either over-sampling or under-sampling methods. In the over-sampling techniques, data samples with minority classes are generated so as to make the majority and minority classes even. The simplest over-sampling technique is done through just duplicating all the minority instances and stacking them up until the data in both minority and majority classes are balanced. This method, however, could easily lead to over-fitting since there is no new information in the training data (Ling and Li 1998; Drummond and Holte 2003). The most common over-sampling approach is recognized to be the Minority Over-Sampling Technique or SMOTE. SMOTE increases the minority instances by randomly generating synthetic data through interpolation between two minority samples

based on the nearest neighbor of the data. Nevertheless, this method may cause over-generalization because SMOTE generates the minority data without considering the majority data (Yen and Lee 2009). Moreover, all the over-sampling techniques increase the size of the training data, and thus may lead to much more computational time (Liu et al. 2011). Some examples of variants on SMOTE are SMOTE and Tomek Links (Batista et al. 2004), SMOTE and SVM ensembles (Liu et al. 2011) and SMOTE based on clustering (Yong 2012).

The under-sampling approaches balance the data by reducing the number of data samples within the majority class so as to reduce the degree of imbalanced data distribution. The simplest under-sampling method is known as the random under-sampling approach or “RT” (Yen and Lee 2009), where the training data is balanced by randomly selecting a subset of the majority instances and then combining them with the minority instances. In this case, the size of data samples in both minority and majority classes must be the same. Other techniques based on the under-sampling approaches are sampling based on distance (Chyi 2003; Zhang and Mani 2003). In Chyi (2003), a subset of the samples with the majority class is chosen based on the “nearest”, “farthest”, “average nearest”, and “average farthest” distances between the majority and minority samples. However, in practice, this method is not efficient since it takes too much computationally effort in selecting the samples with the majority class. A similar method is presented by Zhang and Mani (2003). Their results show that their method and RT perform equally well. Using the under-sampling approaches, nonetheless, might delete crucial data and some valuable information might be lost (Akbani et al. 2004).

Regarding the algorithmic approaches, there are some works attempting to solve imbalanced data problems. These include SVM with cost sensitive learning (Elkan 2001; Drummond and Holte 2003; Sun et al. 2007) and SVM with the quadratic cost function, recognized as weighted Lagrangian SVM (Hwang et al. 2011). Here the cost-sensitive learning assumes high misclassification cost for the samples in the minority class and thus tries to minimize the high cost of errors. Some research has used combinations of both data and algorithm approaches. Tang et al. (2002) used the over and under-sampling techniques along with cost-sensitive learning for SVM. Liu et al. (2011) formed a committee of SVM using GA in conjunction with over and under-sampling techniques. Ren (2012) balances the data using the random over-sampling method with the proposed optimal threshold for decision boundaries.

As already mentioned, in vector quantization (VQ) technique, data are represented by means of prototypes. VQ approaches, either supervised or unsupervised,

conceptually work by dividing a large set of vectors into groups called Voronoi cells, and each group is then represented by its centroid or prototype. Therefore, the goal of prototype based method is to represent data by means of prototype vectors such that they represent the data distribution as accurately as possible. The learning vector quantization (LVQ) as an important neural model in supervised vector quantization can be associated to SMOTE (Nakamura et al. 2013). Since the existing SMOTE algorithms have some drawbacks such as identifying the proper borderlines between classes, LVQ based SMOTE as an over-sampling method can defeat this limitation by generating synthetic samples which occupy more feature space than the other SMOTE algorithms. In addition, this method generates synthetic samples using real samples taken from reference datasets according to a similarity measure of codewords such that the set of all the codewords, i.e. a codebook, is obtained by LVQ. Young et al. (2015) proposed an over-sampling technique called V-synth. This method relies on the properties of Voronoi cells to generate useful synthetic minority points. The main idea of using V-synth technique is to identify the particular receptive fields in feature space which is appropriate to generate synthetic minority samples.

Proposed framework

GRSOM algorithm for growing new data

Originally, GRSOM (Sasamura et al. 2002) had been successfully used in the travelling salesman’s problem (TSP) in order to find the shortest route under the condition that every city need to be visited once with minimum total distance. Classical GRSOM consists of input and output layers. In the input layer, each of the input nodes corresponds to each of the 2-dimensional synaptic vectors located in between cities. In the output layer, the outputs correspond to cells which will be included into ring topology. Note that each cell has an associated synaptic vector in input layer. According to Bai et al. (2006a, b), the evolution of the GRSOM network can be imagined as the stretching of a ring toward all the cities to be visited. Once the nodes are constructed based on the minimum Euclidian distance between the cities and the node themselves, the tour route is formed via connecting each of the cities to the nearby nodes.

In the context of the data balancing technique, GRSOM will be slightly modified and used to balance the data through growing the new data to some extent so as to make the majority and minority data even. Here, each of the input nodes corresponds to each of the data samples used for a classification problem, in which, these samples are

derived from a combination of the original data samples and the new grown instances. As before, each of the output nodes corresponds to each of the cells in ring topology. Once the GRSOM network is evolved, the new nodes are created and inserted into the ring. Similarly to the TSP applications, the new nodes or new grown instances are created in such a way that topology of the tour route or of the data is preserved.

In this work, we apply GRSOM to imbalanced data sets in which class labels are available. GRSOM structure consists of two major modules: the input space and the ring topology space. At input space, prototype vectors w_j are used to approximate a distribution of the input vectors x_i , $i = 1, \dots, s$. On the topology space, a connectivity between the neurons is arranged on a ring structure such that each neuron is connected to both sides of neurons. A prototype vector called weight vector is associated to each neuron. This feature is used to connect both modules together and also to preserve topological ordering and a neighborhood structure of the prototypes. Let $[\mathbf{X}; \mathbf{Y}]$ be an original training data set for a binary classification problem, such that $\mathbf{X}$ and $\mathbf{Y}$ are written in the following form:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{s,1} & x_{s,2} & \cdots & x_{s,d} \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_s \end{bmatrix}, \quad (1)$$

where $y_i \in \{0, 1\}$ for $i = 1, \dots, s$. Each training pair (x_i, y_i) consists of an attribute vector $x_i \in \mathbb{R}^d$ and a class label $y_i \in \mathbb{R}^1$, where $y_i = 0$, if the data belongs to class I and $y_i = 1$, if the data belongs to class II. Note, however, that class labels are not used to train GRSOM. Let $\mathbf{X}^+$ be the set which contains the new grown data and let $N(t)$ be the number of grown samples in epoch t . At $t = 0$, GRSOM initializes the prototype vectors w_j , $j = 1, \dots, N(0)$, where $N(0) = 3$, around the center of the input vectors. The ring graph can then be obtained by connecting each prototype to its neighbours. Afterwards, a new set of prototypes which can be considered as the new grown data is included into $\mathbf{X}^+$, i.e. $\mathbf{X}^+ \leftarrow \mathbf{X}^+ \cup \{w_1, w_2, w_3\}$.

At the start of each epoch t , we introduce variables c_j for $j = 1, \dots, N(t)$ which are used as the signal counters such that $c_j = 0$ for all j . Each prototype w_j has an associated counter c_j indicating the frequency of being selected as a winning prototype. The training process for a GRSOM starts with feeding an input vector x_i into the network. Subsequently, GRSOM uses a Euclidean metric in the input space to determine the closeness of an input vector to each prototype w_j . The prototype located closest to the input vector is selected as the winning prototype w_{j^*} , $j^* = \arg \min_{\forall j} \|x_i - w_j\|$. Then, the counter of the winning

prototype is updated by $c_{j^*} \leftarrow c_{j^*} + 1$. Next, the location of each prototype must be changed by using an update rule $w_j \leftarrow w_j + \alpha_t \cdot f(\sigma_t, d(j, j^*)) \cdot [x_i - w_j]$ where the learning rate α_t is a decreasing function of time between $[0, 1]$.

$f(\sigma_t, d(j, j^*)) = \exp\left(-\frac{d_{jj^*}^2}{2\sigma_t^2}\right)$ is a Gaussian shaped curve with neighbourhood range $\sigma_t > 0$, where $d_{jj^*} = \min\{\|w_j - w_{j^*}\|, N(t) - \|w_j - w_{j^*}\|\}$ is the cardinal distance measured along the ring between nodes j and j^* , and $\|\cdot\|$ denotes the Euclidean vector norm. At the end of each epoch, w_p is determined as the prototype with the largest counter value. Among the neighbours of w_p , we set the prototype w_q which is farthest from w_p . Afterwards, we can insert a new prototype w_g , i.e. a new instance, halfway between w_p and w_q according to $w_g = (w_p + w_q)/2$. The counter values of w_p and w_g are updated by $c_p \leftarrow 0.5c_p$ and $c_g \leftarrow 0.5c_p$, respectively. A new grown sample w_g is then included into $\mathbf{X}^+$ which can be written as $\mathbf{X}^+ \leftarrow \mathbf{X}^+ \cup \{w_g\}$. This way, the current number of prototypes in $\mathbf{X}^+$ is updated by $N(t) \leftarrow N(t) + 1$. This process is repeated until the maximum number of training epochs $t_{\max}$ is reached. For convenience, Table 1 provides a step-by-step description of our approach.

Data balancing and model construction

As already mentioned, the proposed framework consists of GRSOM-based data balancing and model construction phases, as depicted in Fig. 1. In the data balancing phase, two proposed algorithms based on the GRSOM method, namely GRSOMO and GRSOMU, are used to balance the samples with the minority and majority classes, respectively. If the over-sampling approach is desired, data balancing is done through the GRSOMO algorithm. If the under-sampling approach is preferred, the GRSOMU algorithm will be used to balance the data. The GRSOMO and GRSOMU algorithms can be described as follows. Let $\mathbf{T}$ be the original training set consisting of the majority samples ($\mathbf{N}$) and the minority samples ($\mathbf{P}$). There are n data samples in the majority class and m samples in the minority class, respectively. For the GRSOMO algorithm, the new grown data set ($\mathbf{X}^+$) is generated by using the minority samples ($\mathbf{P}$) as input for the function $\text{GRSOM}(\cdot, t_{\max})$. The number of new grown data is computed from the difference between the number of the majority and the minority classes, i.e. $n - m$, leading to $t_{\max} \leftarrow n - m - N(t = 0)$. After training with GRSOMO, the new grown data of the minority class are contained in the set $\mathbf{P}^+$. Because of a combination of the original $\mathbf{P}$ with m samples and the new grown data $\mathbf{P}^+$ with $n - m$ samples, we obtain the set $\mathbf{P}^{++}$ consisting of n samples of the minority class from

Table 1 GRSOM algorithm for growing new data

Input: the input data set ($\mathbf{X}$) consisting of s samples with d dimensions, a maximum number of training epochs ($t_{\max}$)	
Output: the new grown data set ($\mathbf{X}^+$)	
i)	BEGIN
ii)	Set $t = 0$, $N(t = 0) = 3$, $\mathbf{X}^+ = \emptyset$.
iii)	Initialize the prototypes $w_j \in \mathbb{R}^d$ using $w_j = \frac{\sum_{i=1}^s x_i}{s} + v_j$, $j = 1, 2, 3$ and $v_j : U(0, 0.1)$.
iv)	$\mathbf{X}^+ \leftarrow \mathbf{X}^+ \cup \{w_1, w_2, w_3\}$.
v)	WHILE $t < t_{\max}$
vi)	Determine the counter value of each prototype $c_j = 0$, $j = 1, K, N(t)$.
vii)	FOR $i = 1$ to s
viii)	Randomly select $x_i \in \mathbb{R}^d$ from $\mathbf{X}$ without replacement.
ix)	Find the winning neuron j^* corresponding to the prototype vector (w_{j^*}) closest to the input vector x_i , such that $j^* = \arg \min_{j \in \{1, \dots, N(t)\}} \ x_i - w_j\ $.
x)	Assign counter value for each prototype using $c_j = \begin{cases} c_j + 1, & \text{if } j = j^* \\ c_j, & \text{otherwise} \end{cases}, j = 1, 2, \dots, N(t).$
xi)	Update all prototypes, i.e. w_j , $j = 1, 2, \dots, N(t)$, using the recursive formula: $w_j \leftarrow w_j + \alpha_t \cdot f(\sigma_t, j, j^*) \cdot [x_i - w_j]$ where the learning rate α_t is a decreasing function of time, and $f(\sigma_t, j, j^*)$ is a neighborhood function defined by $f(\sigma_t, j, j^*) = \exp \left(\frac{-\ r_j - r_{j^*}\ ^2}{2\sigma_t^2} \right).$ Note that r_j and r_{j^*} are the locations of neurons j and j^* in the one-dimensional ring topology, respectively. The function σ_t is an exponentially decreasing function.
xii)	END FOR
xiii)	Find w_p corresponding to the maximum value of the signal counter.
xiv)	Find w_q using $w_q = \begin{cases} w_{p-1} & \text{if } \ w_{p-1} - w_p\ \geq \ w_{p+1} - w_p\ \\ w_{p+1} & \text{otherwise.} \end{cases}$
xv)	Insert a new grown sample between w_p and w_q which can be written as $w_g = 0.5(w_p + w_q).$
xvi)	Re-assign the counter values of w_p and w_q as $c_p \leftarrow 0.5c_p$ and $c_g \leftarrow 0.5c_p$.

$\mathbf{P}^{++} \leftarrow \mathbf{P} \cup \mathbf{P}^+$. This way, the number of samples in the minority class will be equal to the number of samples in the majority class. Then, the updated minority samples $\mathbf{P}^{++}$ in combination with the original majority samples $\mathbf{N}$ can be considered as the balanced training dataset $\mathbf{T}^+$, i.e. $\mathbf{T}^+ \leftarrow \mathbf{P}^{++} \cup \mathbf{N}$. Conversely, if GRSOMU is desired, the majority samples ($\mathbf{N}$) are used as input for the function GRSOM ($\cdot, t_{\max}$). The number of new grown data needs to

be equal to the number of the minority samples. By using the under-sampling approach, the new grown data of the majority class is represented by $\mathbf{N}^+$ with m samples, in which, $t_{\max} \leftarrow m - N(t = 0)$. As before, the balanced training dataset $\mathbf{T}^+$ can be considered as the combination of the new majority samples $\mathbf{N}^+$ and the original minority samples $\mathbf{P}$. The algorithmic description of GRSOMO and GRSOMU methods can be found in Table 2.

Table 1 continued

xvii)	Increase the number of prototypes which can be considered as the number of samples: $N(t) \leftarrow N(t) + 1.$
xviii)	$t \leftarrow t + 1.$
xix)	$\mathbf{X}^+ \leftarrow \mathbf{X}^+ \cup \{w_g\}.$
xx)	END WHILE
xxi)	return $\mathbf{X}^+.$
xxii)	END

Fig. 1 Proposed framework diagram

In the model construction phase, either neural networks with the backpropagation algorithm (BPN) or support vector machines (SVM) are used as a classifier. The bootstrap algorithm is used to generate training samples for several classifiers, and these classifiers will be used to form a committee network. A potential reason for this is that we want to encourage each of the classifiers (committee members) to learn different parts of data so that they all could have different expertise and help one another to perform a classification task. Finally, the decisions made

by each of the committee members are combined using the majority voting scheme to produce the committee decision. Table 3 shows the bootstrap committee algorithm.

Experimental method

The following describes experimental settings used to evaluate the proposed models' performance under various levels of imbalanced data.

Table 2 The GRSOMO and GRSOMU algorithms

-
- (i) Let T be the original unbalanced training set with $n + m$ samples, where $n > m$. And let $T = P \cup N$, where P contains only the data in the minority class with m samples and N contains only data in the majority class with n samples
- (ii) IF GRSOMO is desired THEN DO steps (iii)–(v)
- (iii) Use GRSOM to generate new data from P for $n - m$ samples such that P is used as an input of the GRSOM function, i.e. $\text{GRSOM}(\mathbf{X}, t_{\max})$ where $\mathbf{X} \leftarrow P$. Then, the function will return the $n - m$ samples which are contained in the new grown data set $\mathbf{X}^+$, i.e. $\mathbf{X}^+ \leftarrow \text{GRSOM}(\mathbf{X}, t_{\max})$. Note that, in the case of over-sampling approach, $t_{\max} \leftarrow n - m - N(t = 0)$
- (iv) $P^+ \leftarrow \mathbf{X}^+$ and $P^{++} \leftarrow P \cup P^+$. Thus, the number of samples in the minority class can be adjusted from m to $m + (n - m) = n$ samples which equals to the number of samples in the majority class
- (v) Define the balanced training set as $T^+ \leftarrow P^{++} \cup N$ and GO TO ix)
- (vi) IF GRSOMU is desired THEN DO steps vii) to viii)
- (vii) Use GRSOM to generate new data from N for m samples in which N is used as an input of the GRSOM function, i.e. $\text{GRSOM}(\mathbf{X})$ where $\mathbf{X} \leftarrow N$. Then, the function will return the new grown data set $\mathbf{X}^+$ with m samples, i.e. $t_{\max} \leftarrow m - N(t = 0)$, $\mathbf{X}^+ \leftarrow \text{GRSOM}(\mathbf{X}, t_{\max})$
- (viii) $N^+ \leftarrow \mathbf{X}^+$ and $T^+ \leftarrow N^+ \cup P$. Note that only the m samples will be generated for the majority class which equals to original number of samples in the minority class. This leads to the balanced training set T^+
- (ix) Return T^+
- (x) END
-

Table 3 Bootstrap committee network construction (adopted and modified from Chetchotsak and Twomey 2007)

-
- (i) Let $T^+ \leftarrow A \cup B$, where A belongs to class I and B to class II
- (ii) Let $\hat{F}$ be the empirical probability distribution where A and B each with u observations are drawn
- (iii) Let $a_1, a_2, \dots, a_u$ be a collection of A and $b_1, b_2, \dots, b_u$ be a collection of B , where $a_i = (x_i, y_i = 0)$ and $b_i = (x_i, y_i = 1)$, $i = 1, 2, \dots, u$
- (iv) Specify the number of bootstrap samples, r to produce A_j^* and B_j^* , $j = 1, 2, \dots, r$
- (v) For each of A_j^* and B_j^* , a_i is randomly chosen from A and b_i from B for $i = 1, 2, \dots, u$ with replacement and equal probability mass $\frac{1}{u}$
- (vi) Repeat step (v) r times to construct A_j^* and B_j^* , $j = 1, 2, \dots, r$
- (vii) $T_j^* \leftarrow A_j^* \cup B_j^*$, $j = 1, 2, \dots, r$
- (viii) Train each of the committee members using T_j^* , where $j = 1, 2, \dots, r$
- (ix) Form a committee network using majority voting scheme
- (x) END
-

Model settings and baseline methods

To evaluate the effectiveness of our proposed data balancing methods, we compare the performance of GRSOMO and GRSOMU to that of SMOTE and RT. These data balancing methods will be used in conjunction with either BPN or SVM so as to construct a committee model. Moreover, a single BPN and SVM are also used to compare against the committees of these classifiers. In this paper, single classifiers constructed using GRSOMO, GRSOMU, RT, and SMOTE are recognized as “GRSOMO”, “GRSOMU”, “RT”, and “SMOTE”, respectively. The terms “CnGRSOMO”, “CnGRSOMU”, “CnRT”, and “CnSMOTE” symbolize the committee networks constructed using those corresponding methods. In this case, a model that was built using an imbalanced data set is referred to as “Original”. Table 4 describes parameter settings for the classifiers and GRSOM algorithms. Here the number of bootstrapped committee members should be an odd number due to the majority

Table 4 Parameter settings for the classifiers and GRSOM algorithm

Algorithms	Parameters
GRSOM	$\alpha_t = \frac{0.01}{1+0.01(t-1)}$ $f(\sigma_t, d(h, k)) = \exp\left(\frac{-\ r_h - r_k\ ^2}{2\sigma_t^2}\right)$ $\sigma_t = 0.4 \left(\frac{10^{-9}}{0.4}\right)^{t/1000}$ $t_{\max} = n - m - N(t = 0)$, for GRSOMO $t_{\max} = m - N(t = 0)$, for GRSOMU
BPN	Learning cycles = 50,000 Hidden units = 20
SVM	Kernel function = RBF
Committee networks	Number of bootstrap committee members = 31 Fusion rule = Majority vote

voting method and the number of neural networks to form a committees being required to be at least 20–30 (Parmanto et al. 1996).

Table 5 Description of the UCI data sets

Data sets	Sample sizes	Numbers of attributes	Majority class ($y_i = 1$)	Minority class ($y_i = 0$)	Imbalance levels (ratio*)
Ecoli (im)	336	7	Class $\neq$ im	Class = im	3.36:1
Ecoli (imU)	336	7	Class $\neq$ imU	Class = imU	8.6:1
Abalone (9 vs. 18)	731	8	Class $\neq$ 9	Class = 18	16.40:1
Yeast (ME2)	1484	8	Class $\neq$ ME2	Class = ME2	28.1:1

* This ratio represents the ratio of majority instances to minority instances

Data sets

The data sets used in this study are from UCI Machine Learning Repository, Center for Machine Learning and Intelligent Systems. These data sets are described in Table 5. The imbalance ratios of the data sets range from 3.36:1 to 28.1:1.

Performance measure

To measure performance of the classification models, five-fold cross validation is used in this study. For consistency, the samples with the minority class are referred to as “positive” and the samples with the majority class as “negative”. Table 6 illustrates the confusion matrix for a binary class problem.

TP and TN denote correctly classified positive and negative samples while FP and FN mean falsely classified positive and negative samples, respectively. According to Yen and Lee (2009), the following are used to measure the classification performance for the minority class:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \text{ and} \quad (3)$$

$$MI's F - \text{measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{Precision} + \text{recall}}. \quad (4)$$

In this regard, precision measures how well a classifier performs, given that the classifier predicts “positive” whereas recall measures how well the classifier performs, given that the samples are actually “positive”. Generally, precision and recall are trade-offs; i.e., if the classifier has a high precision rate, the recall rate will be low. As a result,

Table 6 The confusion matrix

	Predicted positive	Predicted negative
Actual positive	TP	FN
Actual negative	FP	TN

MI's F-measure which compromises both recall and precision is also used to evaluate the classification performance.

Experimental trials

To remove dependency on sampling of training data, the experiment is replicated for, say, 10 times. This is done by sampling the training set for each data problem 10 times and each time all the classifiers are trained and validated through five-fold cross validation, according to the experimental settings. Then the performance measures of each model can be computed using the average of such measures over ten trials.

Results and discussion

Experimental results are reported in terms of the average values of precision, recall and MI's F-measure over ten trials. These results are presented as follows.

Comparison among data balancing techniques

Figures 2, 3, 4 and 5 show the results for the four data problems using MI's F-measures. In general, the classifiers constructed using balanced data perform much better than those with imbalanced data. When the imbalance ratio becomes higher (Figs. 3, 4, 5), all the over-sampling techniques seem to outperform the under-sampling methods. Such results are consistent with those reported in Batista et al. (2004) and Liu et al. (2011). Part of the reason may be because the under sampling technique may delete some important information from the data and hence the classifiers may not learn the data correctly. In this case, the under-sampling techniques rely on the artificially created data while majority of actual data have been substituted.

Figures 2, 3, 4 and 5 reveal that the classifier constructed using GRSOMO performs at least as well as one built using SMOTE. Such results support our hypothesis that GRSOM can balance the data more effectively than SMOTE since it grows new minority data while preserving

Fig. 2 MI's F-measure for Ecoli (im) problem: imbalance ratio 3.36

Fig. 3 MI's F-measure for Ecoli (imU) problem: imbalance ratio 8.19

Fig. 4 MI's F-measure for abalone (9 vs. 18) problem: imbalance ratio 16.81

the topology of data. Note that, GRSOM inserts a new prototype through interpolation between two prototype vectors such that both vectors are the prototype with the largest counter value and its neighbor, respectively, while

Fig. 5 MI's F-measure for yeast (ME2) problem: imbalance ratio 28.1

SMOTE generates new data through interpolation between two random samples. Thus, SMOTE may induce noise into the new balanced data. In addition, SMOTE is categorized as an over-sampling approach, while GRSOM is applied to both over-sampling and under-sampling schemes.

Comparison among learning algorithms

Figure 6 provides a 95 % confidence interval (95 % C.I.) plot of the MI's F-measure for all the classifiers using the over-sampling techniques. A robust classifier should have a high average value of MI's F-measure with a small confidence band. It is obvious that the committee models outperform the single models in all cases. Such results concur with those reported in most literature. This is because all the classifiers in the committee models are encouraged to learn different parts of data so as to have different expertise and help one another to perform a classification task. Here, CnGRSOMO which is constructed using GRSOM techniques and formed through the committee of BPN seems to be the most robust in this study. CnGRSOMO performs as well as or better than CnSMOTE for all imbalance ratios.

Regarding learning algorithms for the classifiers, the models constructed based on BPN seems to perform better than those based on SVM in most cases. The results are quite obvious when imbalance ratios become large (Figs. 3, 4, 5). Such findings agree with those reported in Zhang et al. (2012). However, comprehensive investigation of those findings is still needed for the sake of clarification.

MI's F-measure is then decomposed into precision and recall as depicted in Figs. 7 and 8. A good classifier should have high MI's F-measure and thus be able to balance both precision and recall. For most cases as in Fig. 7, if recall is high then precision is low. CnGRSOMO with SVM for instance has higher recall but much smaller precision rates compared to CnGRSOMO with BPN. As a result, it has a

Fig. 6 MI's F-measure for BPN and SVM models with the over-sampling methods

Fig. 7 The experimental result for Abalone (9 vs. 18) problem: imbalance ratio 16.81

Fig. 8 The experimental result for yeast (ME2) problem: imbalance ratio 28.1

substantially smaller MI's F-measure than CnGRSOMO of the BPN based model. In this case, CnGRSOMO with BPN has the highest MI's F-measure and therefore it can effectively balance both precision and recall values. Such occurrence is also valid when imbalance ratios become larger as depicted in Fig. 8. CnGRSOMO of the BPN model is hence most robust in this study. Nevertheless, further intensive investigation should be conducted to explain such incidence.

Computational expense

This section discusses the computational expenses of each method. Computational time in this study can be broken into two parts. The first part is for data balancing while the second part is for construction of classifiers. Table 7 shows computation complexity of the data balancing techniques, where the symbol " $\mathcal{O}(\dots)$ " represents degree of complexity. Here, the most time-consuming technique is

Table 7 Computational complexity of data balancing methods

Balancing techniques	Computational complexity
GRSOMO	$\mathcal{O}(m^3 + n^2)$
GRSOMU	$\mathcal{O}(m^2n)$
SMOTE	$\mathcal{O}(m^3)$
RT	$\mathcal{O}(m)n$

GRSOMO while the quickest computation time belongs to RT.

For classifier construction, it is quite clear that the committee models are more computationally exhaustive than the single models. In this regard, the computation time required to construct a committee of classifiers is roughly equal to r times that required to train a single classifier. Furthermore, it is generally known that training a BPN model requires much more time than training an SVM model. In this study CnGRSOMO with BPN is the most time consuming method.

Generalization of results and future direction

Our proposed methods have been tested using four real data problems with imbalance ratios ranging from 3.36:1 to 28.1:1, and the number of attributes between 7 and 8. As a result, there is enough evidence to believe that our proposed method, CnGRSOMO with BPN, can perform at least as well as CnSMOTE. However, in this study, the use of our proposed method is limited to binary classification problems. It is unclear whether our method can be used with a multi-classification problem. In addition, the proposed method should be used in conjunction with some dimension reduction techniques in order to reduce computational time. Therefore, our future direction is to develop an algorithm based on GRSOM to remove such limitations.

Conclusion

This paper introduces a new technique to improve classification performance for imbalanced data problems. Our results suggest that the best of our proposed methods, CnGRSOMO with BPN, is the most robust method. In this technique, GRSOMO is used to balance the data and then a committee of classifiers based on BPN is constructed to perform a classification task. The results reveal that CnGRSOMO with BPN can perform at least as well as the baseline method for all selected data problems across all imbalance ratios. Moreover, we have found that BPN is more robust than SVM for most imbalanced data cases.

Acknowledgments The first and second authors would like to acknowledge the financial support from the following agencies: NECTEC of NSTDA, I/U CRC in HDD Components, and the Faculty of Engineering, Khon Kaen University, Thailand. The third author was supported by the Thailand Research Fund (TRF), the Office of the Higher Education Commission (OHEC), Khon Kaen University (Grant Number MRG5580032). This research is partially supported by the Centre of Excellence in Mathematics, the Commission on Higher Education, Thailand. Finally, we all would like to thank God for his grace.

References

- Adrianto I, Richman MB, Trafalis TB (2010) Machine learning techniques for imbalanced data: an application for tornado detection. In: Proceedings of the international conference on artificial neural networks in engineering, pp 509–516
- Akbani R, Kwek S, Japkowicz N (2004) Applying support vector machines to imbalanced datasets. In: European Conference on Machine Learning, pp 39–50
- Arnonkijpanich B, Hasenfuss A, Hammer B (2010) Local matrix learning in clustering and applications for manifold visualization. *Neural Netw* 23:476–486
- Arnonkijpanich B, Hasenfuss A, Hammer B (2011) Local matrix adaptation in topographic neural maps. *Neurocomputing* 74:522–539
- Bai Y, Zhang W, Hu H (2006a) An efficient growing ring SOM and its application to TSP. In: Proceedings of the international conference on applied mathematics, pp 351–355
- Bai Y, Zhang W, Jin Z (2006b) An new self-organizing maps strategy for solving the traveling salesman problem. *Chaos Solitons Fract* 28:1082–1089
- Batista A, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor* 6:20–29
- Chan PK, Wei F, Prodromidis A, Stolfo SJ (1999) Distributed data mining in credit card fraud detection. *IEEE Intell Syst* 14:67–74
- Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: Synthetic Minority Over-sampling Technique. *J Artif Intell Res* 16:321–357
- Chawla NV, Lazarevic A, Hall LO, Bowyer KW (2003) Smoteboost: improving prediction of the minority class in boosting. In: Proceedings of the European conference on principles and practice of knowledge discovery in databases, pp 107–119
- Chetchotsak D, Pattanapairoj S (2010) Committee network model for HDD functional tests. In: Proceedings of international conference on artificial neural networks in engineering, pp 629–636
- Chetchotsak D, Twomey JM (2007) Combining neural networks for function approximation under conditions of sparse data: the biased regression approach. *Int J Gen Syst* 36:479–499
- Chyi YM (2003) Classification analysis techniques for skewed class distribution problems. Master thesis, Department of Information Management, National Sun Yat-Sen University
- Cohen G, Hilario M, Sax H, Hugonnet S, Geissbuhler A (2006) Learning from imbalanced data in surveillance of nosocomial infection. *Artif Intell Med* 37:7–18
- Daskalaki S, Kopanas I, Avouris N (2006) Evaluation of classifiers for an uneven class distribution problem. *Appl Artif Intell* 20:381–417
- Drummond C, Holte RC (2003) C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: Proceedings of the international conference on machine learning

- Elkan C (2001) The foundations of cost-sensitive learning. In: Proceedings of international joint conference on artificial intelligence, pp 973–978
- Fawcett T, Provost F (1997) Adaptive fraud detection. *Data Min Knowl Discov* 1:291–316
- Fernandez A, Garcia S, Jesus MJ, Herrera F (2008) A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets Syst* 159:2378–2398
- Ganji MF, Abadeh MS, Hedayati M, Bakhtiari N (2010) Fuzzy classification of imbalanced data sets for medical diagnosis. In: Proceedings of Iranian conference on biomedical engineering, pp 1–5
- Hilas CS, Mastorocostas PA (2008) An application of supervised and unsupervised learning approaches to telecommunications fraud detection. *Knowl Based Syst* 21:721–726
- Huang YM, Hung CM, Jiau HC (2006) Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem. *Nonlinear Anal Real World Appl* 7:720–757
- Hwang JP, Park S, Kim E (2011) A new weighted approach to imbalanced data classification problem via support vector machine with quadratic cost function. *Expert Syst Appl* 38:8580–8585
- Kang P, Cho S, MacLachlan DL (2012) Improved response modeling based on clustering, under-sampling, and ensemble. *Expert Syst Appl* 39:6738–6753
- Kubat MR, Holte C, Matwin S (1998) Machine learning for the detection of oil spills in satellite radar images. *Mach Learn* 30:195–215
- Li DC, Liu CW, Hu SC (2010) A learning method for the class imbalance problem with medical data sets. *Comput Biol Med* 40:509–518
- Ling C, Li C (1998) Data mining for direct marketing: problems and solutions. In: Proceedings of international conference on knowledge discovery and data mining, pp 73–79
- Liu Y, Yu X, Huang JX, An A (2011) Combining integrated sampling with SVM ensembles for learning from imbalanced dataset. *Inf Process Manage* 47:617–631
- Mazurowski MA, Habas PA, Zurada JM, Lo JY, Baker JA, Tourassi GD (2008) Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance. *Neural Netw* 21:427–436
- Nakamura M, Kajiwar Y, Otsuka A, Kimura H (2013) LVQ-SMOTE—learning vector quantization based synthetic Minority Over-Sampling Technique for biomedical data. *BioData Min* 6:16
- Nanthapodej R, Chetchotsak D (2009) Classification performance of committee networks improvement under sparse data conditions. *Khon Kaen Univ Res J* 9:65–76
- Parmanto B, Munro PW, Doyle HR (1996) Reducing variance of committee prediction with resampling techniques. *Connect Sci* 8:405–425
- Ren J (2012) ANN vs. SVM: which one performs better in classification of MCCs in mammogram imaging. *Knowl Based Syst* 26:144–153
- Sasamura H, Ohta R, Saito T (2002) A simple learning algorithm for growing ring SOM and its application to TSP. In: Proceedings of international conference on neural information processing, pp 1287–1290
- Sun Y, Kamel MS, Wong A, Wang Y (2007) Cost-sensitive boosting for classification of imbalanced data. *J Pattern Recogn Soc* 40:3358–3378
- Tang Y, Zhang YQ, Chawla NV, Krasser S (2002) SVMs modeling for highly imbalanced classification. *IEEE Trans Syst Man Cybern Part B Cybern* 39:281–288
- Yen SJ, Lee YS (2009) Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Syst Appl* 36:5718–5727
- Yong Y (2012) The research of imbalanced data set of sample sampling method based on k-means cluster and genetic algorithm. *Energy Procedia* 17:164–170
- Young W, Nykl S, Weckman G, Chelberg D (2015) Using Voronoi diagrams to improve classification performances when modeling imbalanced datasets. *Neural Comput Appl* 26:1041–1054
- Zhang J, Mani I (2003) kNN approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings of the ICML workshop on learning from imbalanced dataset
- Zhang Y, Zhang D, Mi G, Ma D, Li G, Guo Y, Li M, Zhu M (2012) Using ensemble methods to deal with imbalanced data in predicting protein–protein interactions. *Comput Biol Chem* 36:36–41