Expert Systems With Applications 98 (2018) 72–83

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

An overlap-sensitive margin classifier for imbalanced and overlapping data

Han Kyu Lee, Seoung Bum Kim*

School of Industrial Management Engineering, Korea University, 145 Anamro, Seongbuk-gu, Seoul 02841, Republic of Korea

ARTICLE INFO

Article history:
Received 16 August 2017
Revised 6 January 2018
Accepted 7 January 2018
Available online 9 January 2018

Keywords:
Classification
Imbalanced class
Overlapping class
Support vector machine

ABSTRACT

Classification is an important task in various areas. In many real-world applications, class imbalance and overlapping problems have been reported as major issues in the application of traditional classification algorithms. An imbalance problem occurs when training data contain considerably more representatives of one class than of other classes. Class overlap occurs when a region in the data space contains a similar number of data for each class. When a class overlap occurs in imbalanced data sets, classification becomes even more complicated. Although various approaches have been proposed to deal separately with class imbalance and overlapping problems, only a few studies have attempted to address both problems simultaneously. In this paper, we propose an overlap-sensitive margin (OSM) classifier based on a modified fuzzy support vector machine and *k*-nearest neighbor algorithm to address imbalanced and overlapping data sets. The main idea of the proposed OSM classifier is to separate the data space into soft- and hard-overlap regions using the modified fuzzy support vector machine algorithm. The separated spaces are then classified using the decision boundaries of the support vector machine and 1-nearest neighbor algorithms. Furthermore, by separating a data set into soft- and hard-overlap regions, one can determine which part of the data is to be examined more closely for classification in real-world situations. Experiments using synthetic and real-world data sets demonstrated that the proposed OSM classifier outperformed existing methods for imbalanced and overlapping situations.

1. Introduction

The purpose of classification is to learn the relationship between a set of input variables and a target variable of interest. It has been used in a variety of applications such as disease diagnosis (Koh & Tan, 2005), event classification (Chan, Fan, Prodromidis, & Stolfo, 1999), fault classification in manufacturing (Deng, Tian, Chen, & Harris, 2016; Rostami, Blue, & Yugma, 2016), and intrusion detection in security monitoring (Ashfaq, Wang, Huang, Abbas, & He, 2017; Chand, Mishra, Krishna, Pilli, & Govil, 2016). Most traditional classification algorithms, such as decision trees, artificial neural networks, and support vector machines (SVM), assume a balanced class distribution (He & Garcia, 2009). However, data sets generated in real-world applications frequently exhibit both class imbalance and overlap.

Imbalance problems describe a situation when the number of observations of one class (the majority class) far exceeds that of the other class (the minority class) (Chawla, Japkowicz, & Drive, 2004), as shown in Fig. 1(a). In this case, because almost all the observations are labeled as the majority class, standard classification algorithms tend to skew toward the majority class and ignore the minority class (Guo, Yin, Dong, Yang, & Zhou, 2008).

This situation occurs in many real applications such as fraud detection (Akbani, Kwek, & Japkowicz, 2004; Burez & Van den Poel, 2009; Gür Ali & Aritürk, 2014), text classification (Chawla et al., 2004), rare disease classification (Dangare & Apte, 2012; Mazurowski et al., 2008; Bae, Wu, & Pan, 2010), and nonconforming product classification (Van & Kang, 2016). Typically, class imbalance problems are addressed by balancing the data set or by modifying the classifier (He & Garcia, 2009; Nekooeimehr & Lai Yuen, 2016; Rivera & Xanthopoulos, 2016). Undersampling and oversampling can be used to balance the data set. However, undersampling, which removes the samples in the majority class, can lead to information loss. Oversampling that duplicates the samples in the minority class can lead to overfitting. The synthetic minority oversampling technique (SMOTE) introduced by Chawla, Bowyer, Hall, and Kegelmeyer (2002) has shown significant success in various applications. However, SMOTE has limitations such as overgeneralization and variance (He & Garcia, 2009).

Modifying the classifier is another way to address the imbalance problem. For example, cost-sensitive learning methods assign different costs to each class. Various studies have investi-

* Corresponding author.
E-mail addresses: hankyulee86@korea.ac.kr (H.K. Lee), sbkim1@korea.ac.kr (S.B. Kim).

https://doi.org/10.1016/j.eswa.2018.01.008
0957-4174/

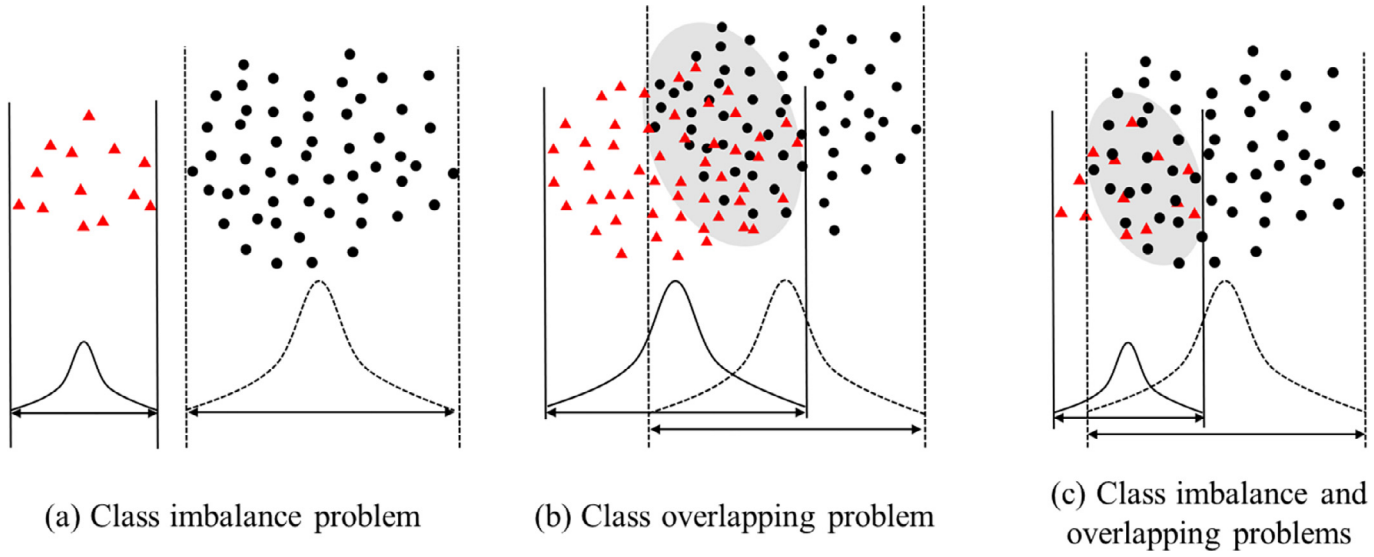


Fig. 1. Challenging situations for classification: (a) class imbalance problem, (b) class overlapping problem, and (c) class imbalance and overlapping problems. The shaded area represents an overlapping region. The solid lines and the dashed lines show the distributions for each class.

gated different types of classifiers and cost-adaptation strategies (Cao, Zhao, & Zaiane, 2013; Elkan, 2001; Kukar & Kononenko, 1998; Napierala & Stefanowski, 2015; Sahin, Bulkan, & Duman, 2013; Turney, 1995; Wang, Gao, Shi, & Wang, 2017). These studies have demonstrated that cost-sensitive learning methods are useful in handling class imbalance problems (Sun, Kamel, Wong, & Wang, 2007; Wang et al., 2017; Zhou & Liu, 2006). Kernel-based SVMs provide robust classification results when applied to imbalanced data sets (Japkowicz & Stephen, 2002). Several SVM-based methods have been proposed to improve classification performance in dealing with imbalanced data sets (Akbani et al., 2004; Batuwita & Palade, 2010; Dai, 2015; Fan, Wang, Li, Gao, & Zha, 2017; Imam, Ting, & Kamruzzaman, 2006; Kang & Cho, 2006; Liu, Yu, Huang, & An, 2011; Wang & Japkowicz, 2010; Yang et al., 2015). Additionally, Zhu and Wang (2017) have introduced an entropy-based matrix learning machine.

Overlapping data contain regions in which the probabilities of the different classes are almost equal (Das, Krishnan, & Cook, 2013), as shown in Fig. 1(b). This situation leads to the development of an inference with almost the same a priori probabilities in this overlapping area, which makes it difficult or even impossible to distinguish between the two classes. In real-world situations, many data sets contain overlapping regions (Tang & Mao, 2010). In such cases, it is difficult for traditional classifiers to find a feasible solution for classification (Xiong, Wu, & Liu, 2010). This is a generically difficult problem for classification tasks. When classifying an overlapping data set, most algorithms perform poorly. Class overlap problems are typically addressed by a kernel function that maps the original data to a highly dimensional feature space (Das et al., 2013; Qu, Su, Guo, & Chu, 2011). This maximizes the extent to which the original data can be separated linearly in a feature space. Nevertheless, class overlap can still exist in this feature space. Some studies have attempted to address overlapping problems with data cleansing approaches, such as Tomek links (Tomek, 1976), the edited nearest neighbor (ENN) rule (Wilson, 1972), SMOTE+ENN, and SMOTE+Tomek (Batista, Prati, & Monard, 2004). However, these data cleansing approaches can result in the loss of important information from the data set. Xiong et al. (2010) introduced simple approaches, including discarding, merging, and separation of data. In their approach, a support vector data description is first applied to find overlapping regions, followed by different classifiers being used for every region. Tang and Mao (2010) suggested using a probabilistic neural network (PNN) algorithm as part of a soft decision strategy to detect overlapping regions. The optimized overlapping regions divide the data set into two parts with low and high confidence relative to the correct classification based on the outcomes from the PNN (Tang & Mao, 2010).

Most existing approaches have been proposed to deal separately with class imbalanced and overlapping problems. However, in numerous real-world applications (e.g., credit card fraud detection, fault detection in semiconductor manufacturing, etc.), these problems occur simultaneously, as shown in Fig. 1(c) (Dorronsoro, Ginel, Sánchez, & Santa Cruz, 1997).

The main purpose of this study is to address the problems in classification caused by both class overlapping and class imbalance. Our proposed method focuses on the separation of data into two regions: the less overlapping and the severely overlapping areas. We then apply different classifiers, appropriate for each region, to maximize classification performance while minimizing the generalization errors under the class imbalance and overlapping situations.

An SVM has been successfully applied to many real-world classification problems in numerous domains (Osuna, Freund, & Girosit, 1997; Tong & Koller, 2001; Wilinski & Osowski, 2009). Based on the structural risk minimization principle, SVMs have shown better generalization performance than other learning algorithms (Vapnik, 1995). Although SVMs are useful classification algorithms, they perform poorly in class imbalanced data sets, similarly to other algorithms (Datta & Das, 2015; Tang, Zhang, & Chawla, 2009). The fuzzy support vector machine (FSVM) is a variant of the SVM algorithm. It assigns different misclassification costs for each observation to reflect its importance (Lin & Wang, 2002). Despite assigning a misclassification cost for each observation, the FSVM, like the SVM method, is still sensitive to the class imbalance problem. To improve classification performance for imbalanced situations, Batuwita and Palade (2010) introduced an FSVM for a class imbalance learning (FSVM-CIL) algorithm. This algorithm can also handle class imbalance problems in the presence of outliers and noise. Although the FSVM-CIL can properly address class imbalance situations, it cannot control class overlap problems.

To overcome this shortcoming, we propose an overlap-sensitive margin (OSM) classifier based on FSVM and *k*-nearest neighbor that can address both class imbalance and overlap problems. The

key points of the OSM classifier assign a new type of misclassification cost (weight) to the FSVM that is different from that used by existing FSVM-based algorithms to address the class imbalance problem. We use a k-nearest neighbor algorithm to measure the degree of overlap of each observation. The trained hyperplane of the OSM classifier separates the training data into hard- and soft-overlap regions. The final decision boundary is obtained according to the different classifiers for each separated region.

The main contributions of this paper can be summarized as follows:

(1) We propose a new weighting scheme in FSVM that can effectively address both the imbalanced and the overlapping data. Other methods including FSVM-CIL can only handle imbalance problems. The proposed weighting scheme uses a nearest neighbor algorithm to quantify the magnitude of overlap for each observation to show the degree of overlap in the data set.
(2) Another key advantage of the proposed OSM classifier is its ability to separate the data into soft- and hard-overlap regions based on the hyperplane of the OSM classifier to achieve higher accuracy in imbalanced and overlapping data. In the soft-overlap region, the observations are classified using the decision boundary of the OSM classifier. In the hard-overlap region, the 1-NN algorithm is used to classify the observations. Although the 1-NN algorithm is an extreme local search algorithm, it shows better classification performance for highly overlapping data (García, Mollineda, & Sánchez, 2008).
(3) To demonstrate the usefulness and applicability of the proposed method, we compared it with other methods using synthetic data from 12 different scenarios and 29 real-world data sets. The results showed that the proposed method outperformed the other methods.

The remainder of this paper is organized as follows. In Section 2, the FSVM algorithm and its variants are briefly reviewed to address the imbalance problem. Section 3 presents the proposed OSM classifier. In Section 4, the experimental results from both synthetic and real-world data sets that were used to examine the properties of the proposed method and to compare its performance with existing methods are described. Finally, Section 5 contains our concluding remarks and suggestions for future work.

2. Related algorithms

This section describes in detail the FSVM-CIL algorithm that combines the FSVM and different error cost (DEC) algorithms (Batuwita & Palade, 2010). The FSVM-CIL is closely related with the proposed OSM algorithm in that both of them modify the weighting scheme of the FSVM to effectively solve class imbalance problems. It is worth noting that the proposed algorithm can address class overlapping problems as well as class imbalance problems.

Lin and Wang (2002) proposed the FSVM, which combines fuzzy logic and an SVM. The FSVM assigns a different fuzzy membership value to each observation based on the importance of the given observation in its class. The FSVM can be trained to consider the different contributions of each observation. The FSVM soft-margin optimization is formulated as follows:

$$Min\left(\frac{1}{2}w\cdot w + C\sum_{i=1}^{k} m_i\xi_i\right)$$
$$s.t.\ y_i(w\cdot\phi(x)+b) \geq 1-\xi_i$$
$$\xi_i \geq 0,\ i=1,\ldots,k, \tag{1}$$

where m_i denotes the membership value that corresponds to observation x_i. If we consider C as the misclassification cost, the different misclassification costs assigned to each observation are denoted as m_iC. Eq. (1) can be transformed into the following dual Lagrangian problem to solve the FSVM optimization problem:

$$MaxW(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}\alpha_i\alpha_j y_i y_j K(x_i,x_j)$$
$$s.t.\ \sum_{i=1}^{m} y_i\alpha_i = 0,\ 0\leq\alpha_i\leq m_iC,\ i=1,\ldots,k. \tag{2}$$

In a dual optimization problem, the upper bound α_i is different for each observation, i.e., by assigning the fuzzy membership value m_i, the FSVM algorithm maximizes the margin by allowing some misclassification according to the importance of the observations. Thus, the FSVM can find the hyperplane that reflects the importance of each observation. The decision function of the FSVM is given in the following equation:

$$f(x) = sign(w\cdot\phi(x_i)+b) = sign\left(\sum_{i=1}^{k}\alpha_i y_i K(x_i,\ x_j)+b\right) \tag{3}$$

Veropoulos, Cristianini, and Campbell (1999) proposed the different error cost (DEC) method, which assigns two misclassification cost values and C^- in the following objective function:

$$Min\left(\frac{1}{2}+w\cdot w + C^+\sum_{[i|y_i=+1]}^{k}\xi_i + C^-\sum_{[i|y_i=-1]}^{k}\xi_i\right)$$
$$s.t.\ y_i(w\cdot\phi(x)+b)\geq 1-\xi_i$$
$$\xi_i\geq 0,\quad i=1,\ldots,k, \tag{4}$$

where C^+ and C^- are the misclassification costs (weights) of the minority (positive) and majority (negative) classes, respectively. To mitigate the effects of class imbalance, the DEC method imposes a higher misclassification cost on minority class observations, i.e., $C^+ > C^-$. When using the DEC method, the decision boundary of the SVM shifts from the minority observations.

The FSVM-CIL method combines the FSVM and DEC methods. By combining the strengths of the two methods, it can assign a higher membership value or higher error cost to the minority class and a lower membership value or lower error cost to the majority class. In addition, this method reflects within-class importance to suppress the effects of outliers and noise, such as the distance from each class center, the distance from the estimated hyperplane, and the distance from the actual hyperplane. The misclassification cost of the minority class observations m_{i^+} and the misclassification cost of the majority class observations m_{i^-} are expressed as follows:

$$m_i{}^+ = f(x_i{}^+)r^+, \tag{5}$$

$$m_i{}^- = f(x_i{}^-)r^-, \tag{6}$$

where r^+ and r^- reflect class imbalances similarly to the DEC method ($r^+ > r^-$). Moreover, the value generated by $f(x_i)$, which is in the range [0, 1], reflects the importance of x_i in its own class. Thus, the FSVM-CIL method can effectively consider the class imbalance problem.

3. The proposed method

Fig. 2 shows an overview of the proposed OSM method. The key feature of the OSM classifier is its weighting scheme, which uses the k-nearest neighbor (kNN) and DEC algorithms to address both class imbalance and overlapping problems. Note that the FSVM-CIL method only addresses class imbalance problems. Another key advantage of the OSM classifier is its ability to separate the data into soft- and hard-overlap regions based on the hyperplane of

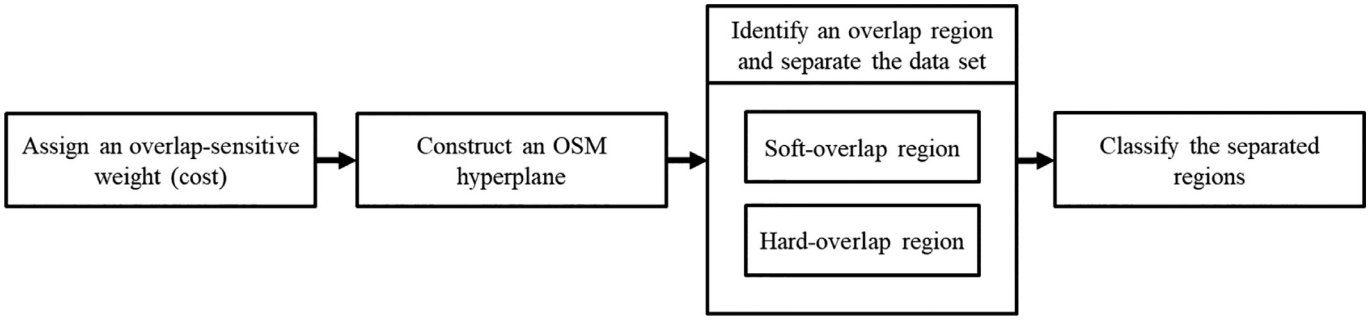


Fig. 2. Overview of the proposed OSM classifier.

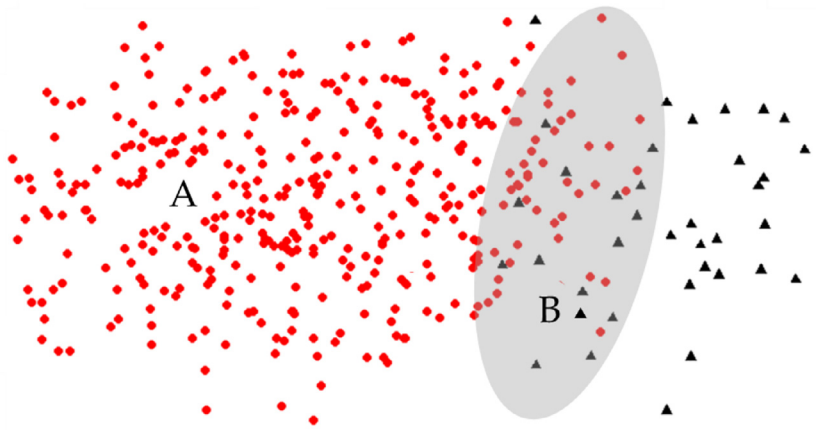


Fig. 3. Imbalanced and overlapping data sets. The red observations are the majority class, whereas the black observations are the minority class. (A) Nonoverlapping region and (B) overlapping region (shaded area).

the OSM classifier. In the soft-overlap region, the observations are classified using the decision boundary of the OSM classifier. In the hard-overlap region, the 1-NN algorithm is used to classify the observations.

3.1. Overlap-sensitive costs

The proposed OSM optimization is formulated as follows (note that the formulation is the same as Eq. (1) except for the overlap-sensitive cost (weight) s_i in the objective function):

$$Min\left(\frac{1}{2}w\cdot w + C\sum_{i=1}^{k} s_i\xi_i\right)$$

$$s.t.\ y_i(w\cdot\phi(x)+b) \geq 1-\xi_i$$

$$\xi_i \geq 0,\ i = 1, \ldots, k. \tag{7}$$

The overlap-sensitive cost s_i is defined as follows:

$$s_i{}^{+} = p(x_i{}^{+})r^{+}, \tag{8}$$

$$s_i{}^{-} = p(x_i{}^{-})r^{-}, \tag{9}$$

where s_i^{+} represents the overlap-sensitive cost of minority class observations x_i^{+}, whereas s_i^{-} represents the overlap-sensitive cost of majority class observations x_i^{-}. r^{+} and r^{-} are values that reflect class imbalance. $p(x_i)$ is the probability indicating the degree of overlap. To address the class imbalance problems, we use the DEC method, i.e., we assign $r^{+}=1$ and $r^{-}=r$, where r is the minority-to-majority class ratio ($r^{+} > r^{-}$). Optimal results for the DEC method can be obtained when C^{-}/C^{+} equals the minority-to-majority class ratio (Akbani et al., 2004). To define function $p(x_i)$ presented in Eqs. (8) and (9), which denotes the degree of overlap in training observations, we propose using the kNN algorithm to measure the degree of overlap for each observation. The kNN algorithm searches only for the k-nearest neighbors of each observation and determines whether the observation belongs to an overlapping or a nonoverlapping region. Fig. 3 shows a hypothetical example of how the kNN algorithm classifies each observation.

We used the 5-nearest neighbors, although different numbers of neighbors k can be used. Later in this paper (Section 4.2), to examine how much different values of k affect the overlap-sensitive costs, we calculated the overlap-sensitive costs with different values of k for real-world data sets. The result showed that k did not significantly affect the overlap-sensitive costs. In Fig. 3, we plotted two arbitrary points: A and B. It can be seen that all 5-nearest neighbors of point A belong to the same class. Consequently, point A is considered to be a nonoverlapping point. In contrast, point B is considered an overlapping point because its 5-nearest neighbors belong to different classes. In this way, we can determine the importance of observations in terms of probability $p(x_i)$ calculated by the kNN algorithm.

If we construct an optimal hyperplane to classify two classes, most support vectors (SVs), which are critical features for constructing a decision boundary in an SVM, are found in the overlapping region. Because a large number of SVs is often a sign of overfitting, we can assume that, if the observations are in the overlap region, they should be assigned a lesser misclassification cost than those assigned to observations in nonoverlapping regions. In other words, observations in a severely overlapping region are less important than those in a nonoverlapping region. Therefore, by using the proposed overlap-sensitive costs, we can find a more robust hyperplane of FSVM that focuses on observations in the nonoverlapping region.

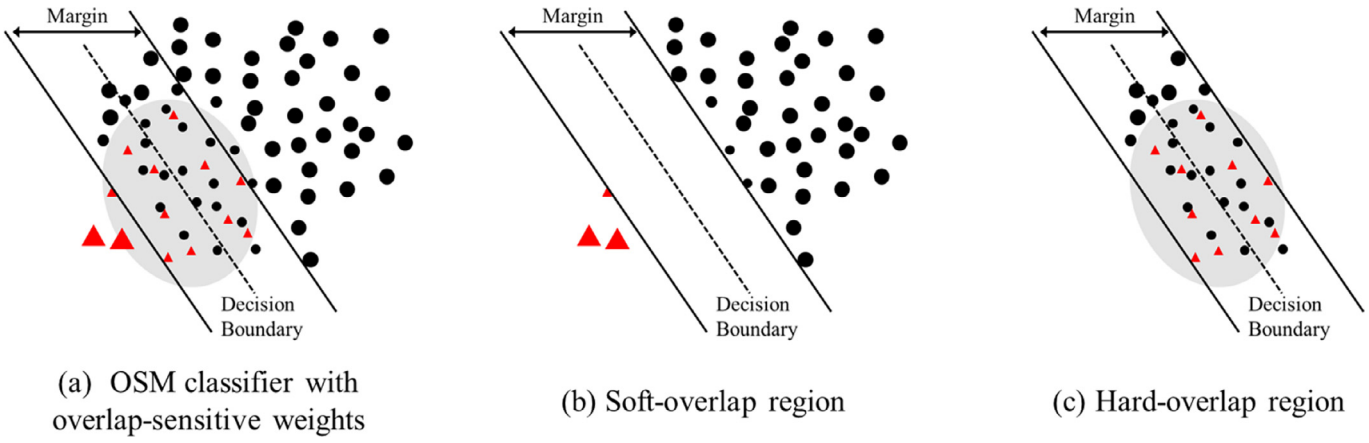


Fig. 4. An OSM classifier trained from two classes. (a) The OSM classifier with overlap-sensitive weights, (b) soft-overlap region, and (c) hard-overlap region. The solid lines represent the maximum margin hyperplanes, whereas the dashed line shows the decision boundary of the OSM classifier. The size of the circles and triangles indicates the degree of overlap-sensitive weights.

3.2. Classification strategy

The proposed OSM classifier is similar to the traditional SVM in that it minimizes the empirical classification errors and maximizes the geometric margin. Two parallel hyperplanes are constructed on each side of the hyperplane that separates the data. This separating hyperplane maximizes the distance between the two parallel hyperplanes. A larger margin between the parallel hyperplanes is assumed to reduce the generalization error of the classifier. Parallel hyperplanes can be described as follows:

$$w \cdot \phi(x) + b = 1 \tag{10}$$

$$w \cdot \phi(x) + b = -1. \tag{11}$$

If the training data are linearly separable, we can select these hyperplanes in such a way that there are no points between them. However, in highly imbalanced and overlapping situations, only a few cases occur in which no points between parallel hyperplanes exist, despite our use of the soft-margin SVM approach to enable separation. Here, we are interested in the parallel hyperplanes, called the margin, to geometrically separate data sets into nonoverlapping and overlapping regions. As illustrated in Fig. 4, we can construct a robust hyperplane by using the OSM classifier with the overlap-sensitive weights for highly imbalanced and overlapping data sets.

In Fig. 4, the triangles and circles represent the minority and majority classes, respectively. The solid and dashed lines show the parallel hyperplanes and the decision boundary, respectively. We consider points in regions with less overlap to be more important; thus, large misclassification costs are assigned to less-overlapped points to reflect their big contributions to the decision boundary of the OSM classifier in Fig. 4(a). The decision boundary is constructed to minimize classification errors for such points in nonoverlapping regions. According to Eqs. (10) and (11), we can easily separate all points in the training data. If the points belong outside the margin, the area is called a soft-overlap region in Fig. 4(b). If the points belong inside the margin, the area is called a hard-overlap region in Fig. 4(c). We propose using different classification algorithms for each region. The observations in the soft-overlap regions are classified by the OSM decision boundary. In the hard-overlap region, the 1-NN algorithm is used to reflect as many local patterns as possible. Although the 1-NN algorithm is an extreme local search algorithm, it shows better classification performance with minority classes than other classifiers for highly imbalanced and overlapping data (García et al., 2008). Consequently, by combining the OSM and 1-NN classifiers, we can achieve a higher classification accuracy in imbalanced and overlapping data.

3.3. Choosing the optimal parameters of the OSM classifier

Our research is based on the use of the margin of the OSM classifier to separate a data space into hard- and soft-overlap regions. However, the OSM classifier can be sensitive to the hyperparameter, such as the trade-off parameter C or the width parameter of the radial basis function kernel. To determine the appropriate hyperparameters, we need to consider two aspects. First, the hard-overlap region should cover observations as compactly as possible to minimize the effects of the 1-NN algorithm. If the hard-overlap region is unnecessarily large, this can lead to overfitting. Second, the OSM classifier should have only a few SVs to avoid high generalization errors. To address these considerations, we used the following two criteria: (1) the proportion of hard-overlap observations (HO) and (2) the proportion of SVs in the training observations (SV):

$$HO(\theta) = \frac{Obs_{hard}}{Obs_{total}}, \tag{12}$$

$$SV(\theta) = \frac{Obs_{sv}}{Obs_{total}}, \tag{13}$$

where θ is the combination of the hyperparameters in the OSM classifier, Obs_{hard} is the number of observations in the hard-overlap region, Obs_{total} is the number of training data, and Obs_{sv} is the number of support vectors. To find the optimal θ, we used the following weighted harmonic mean of HO and SV:

$$\omega_\beta(\theta) = \frac{1}{\frac{\beta}{HO(\theta)} + \frac{(1-\beta)}{SV(\theta)}}, \tag{14}$$

where β is a parameter with a weight in the range [0, 1]. The optimal weight can be obtained so that $\omega_\beta(\theta)$ is minimized:

$$\theta^* = \arg\min_{\theta} \omega_\beta(\theta). \tag{15}$$

4. Experiments and results

4.1. Experiments on synthetic data

We present the experimental results for both synthetic and real-world data sets. To appropriately evaluate classifiers in class imbalance situations, we used the geometric mean (GM) and F1.

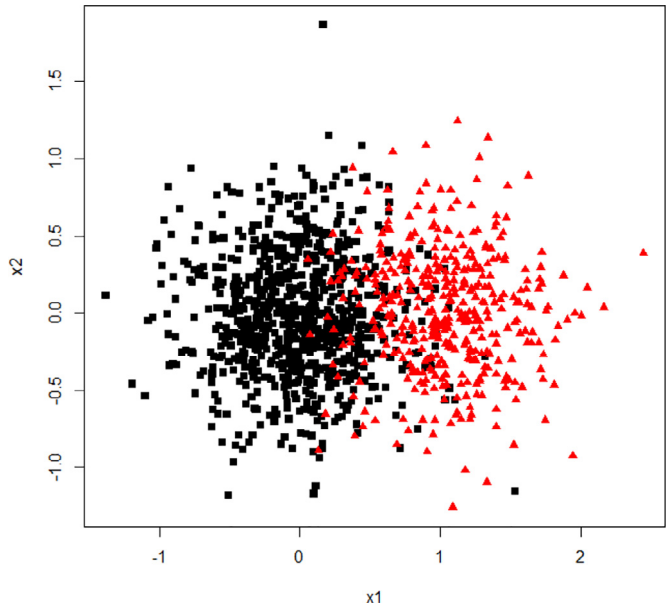


Fig. 5. One thousand observations that considered the class imbalance and overlap situations. These observations were generated and were based on two bivariate normal distributions that had the mean vectors of the majority [0, 0] and minority classes [0, 1].

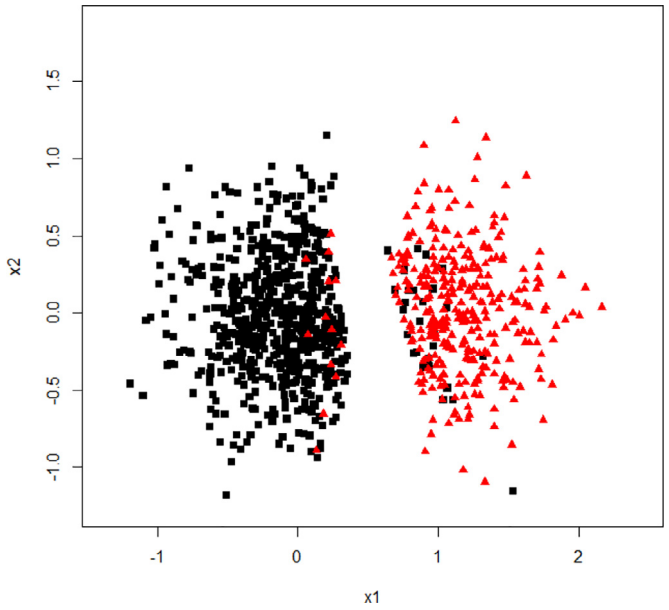


Fig. 6. The observations that belong to the outside margin, which is called the soft-overlap region.

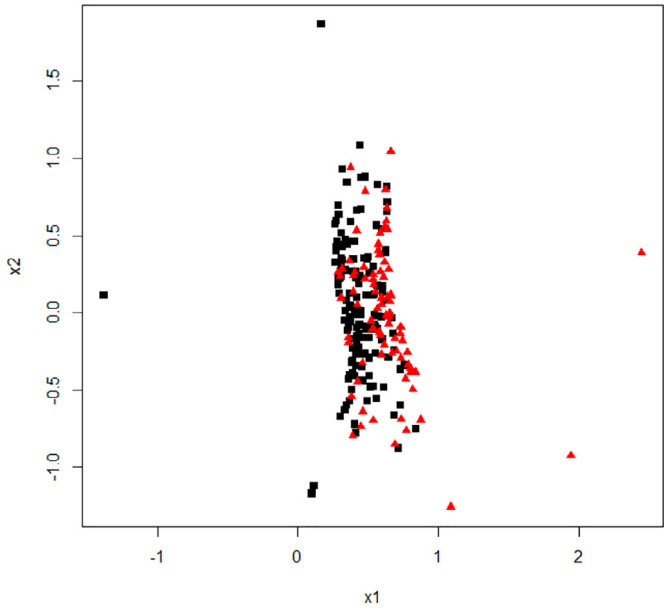


Fig. 7. The observations that belong to the inside margin, which is called the hard-overlap region.

The GM considers the positive and negative classes equally. The GM is calculated as

$$GM = \sqrt{\text{Sensitivity} \times \text{Specificity}} \tag{16}$$

where sensitivity and specificity denote the proportion of positive and negative observations that are classified correctly, respectively (i.e., Sensitivity = TP/(TP+FN), Specificity = TN/(TN+FP)). The F1 is a useful measure that evaluates the detection performance for the target class, which is of particular interest to users. The F1 is calculated as

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{17}$$

where recall is the same as sensitivity and precision is the proportion of correctly predicted observations to all of the predicted observations (i.e., Precision = TP/(TP+FP)).

To clearly understand the properties of the proposed method, we generated 1000 observations that considered the class imbalance and overlap situations. They were based on two bivariate normal distributions that had the mean vector of the majority and minority classes as [0, 0] and [0, 1], respectively, as shown in Fig. 5.

For training the OSM classifier, the kNN algorithm was applied to calculate the degree of overlap of each observation in terms of probability $p(x_i)$. The number of observations, k, was set to 5. A Gaussian kernel was used with width parameter $\gamma = 0.1$ and trade-off parameter $C = 1$. Using Eqs. (10) and (11), we can separate the soft- and hard-overlap regions, as illustrated in Figs. 6 and 7, respectively. We can now identify whether an observation is in the hard-overlap region.

Further experiments were conducted to evaluate the classification performance of the proposed OSM classifier with synthetic data that exhibited various degrees of overlap and imbalance (Table 1).

Twelve data sets with 1000 observations were generated. Each class was drawn from two bivariate normal distributions for both the majority and the minority class. We fixed the majority class mean vector at [0, 0]. Whenever the degree of overlap increased, we gradually set the mean vector of the minority class close to that of the majority class; i.e., Scenario A set the mean vector to [1.3, 1.3], Scenario B set the mean vector to [1, 1], and Scenario C set the mean vector to [0.8, 0.8]. To quantify the degree of overlap, we computed the average overlap-sensitive cost values for the majority and minority classes (fifth and sixth columns in Table 1, respectively). The range of average overlap-sensitive cost was between zero and one. The lower average overlap-sensitive cost values indicated a higher degree of overlap. As shown in the table, the average overlap-sensitive cost values in the majority class were always close to one. Thus, we thought the overlap-sensitive cost values in the minority class could more appropriately represent the degree of overlap.

We randomly split each data set into the training and test data sets (9:1), and the imbalance ratio between the minority and the majority class of both the training and the test data sets was approximately equal as in 1:2, 1:5, 1:10, and 1:20.

Table 1
Different settings for generating synthetic data.

Scenario	Number of observations in the minority class	Number of observations in the majority class	Imbalance ratio	Average overlap-sensitive cost in the majority class	Average overlap-sensitive cost in the minority class	Degree of overlap
A1	333	667	1:2	0.960	0.917	Slight overlap
A2	167	833	1:5	0.985	0.918	
A3	91	909	1:10	0.984	0.837	
A4	48	952	1:20	0.993	0.815	
B1	333	667	1:2	0.928	0.852	Moderate overlap
B2	167	833	1:5	0.958	0.778	
B3	91	909	1:10	0.977	0.760	
B4	48	952	1:20	0.981	0.555	
C1	333	667	1:2	0.869	0.729	Severe overlap
C2	167	833	1:5	0.920	0.585	
C3	91	909	1:10	0.953	0.484	
C4	48	952	1:20	0.976	0.384	

Table 2
Comparative results in terms of the GM and F1 for synthetic data sets. The boldface represents the highest accuracy rates of the GM and F1 for each corresponding scenario.

Data sets	OSM		Normal SVM		Under+SVM		SMOTE+SVM		SDC		Boosting SVM		FSVM-CIL$_{cen}$		FSVM-CIL$_{hyp}$		EFSVM		EMatMHKS		1-NN	
	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1
A1	0.92	0.89	0.35	0.21	0.82	0.77	0.81	0.77	0.81	0.77	0.83	0.80	0.86	0.82	0.87	0.84	0.88	0.84	**0.93**	**0.90**	0.90	0.86
A2	**0.90**	**0.83**	0.28	0.13	0.74	0.60	0.67	0.55	0.67	0.55	0.87	0.76	0.74	0.64	0.75	0.65	0.76	0.67	0.75	0.49	0.82	0.73
A3	0.82	0.74	0.57	0.37	0.81	0.50	0.77	0.59	0.77	0.59	**0.89**	**0.84**	0.71	0.58	0.73	0.59	0.74	0.61	0.74	0.32	0.84	0.74
A4	0.76	**0.68**	0.38	0.16	**0.84**	0.26	0.50	0.25	0.50	0.25	0.55	0.40	0.45	0.29	0.45	0.29	0.45	0.29	0.81	0.20	0.77	0.67
B1	0.78	0.73	0.46	0.33	0.75	0.68	0.75	0.67	0.75	0.67	0.76	0.68	0.78	0.70	0.77	0.70	0.77	0.71	**0.84**	**0.80**	0.78	0.62
B2	0.67	0.48	0.48	0.30	0.70	0.50	0.68	0.51	0.68	0.51	0.60	0.45	**0.72**	**0.57**	0.71	0.56	0.71	0.56	0.61	0.41	0.65	0.53
B3	0.62	**0.51**	0.52	0.28	0.67	0.31	**0.69**	0.39	**0.69**	0.39	0.54	0.39	0.64	0.44	0.66	0.46	0.63	0.44	0.58	0.21	0.60	0.50
B4	**0.72**	**0.51**	0.49	0.16	0.62	0.17	0.52	0.21	0.52	0.21	0.13	0.03	0.30	0.12	0.30	0.12	0.30	0.11	0.66	0.14	0.64	0.40
C1	**0.96**	**0.91**	0.53	0.43	0.91	0.87	0.90	0.87	0.87	0.83	0.87	0.83	0.89	0.86	0.89	0.86	0.89	0.86	0.87	0.62	0.93	0.89
C2	**0.90**	**0.87**	0.24	0.10	0.87	0.55	0.75	0.65	0.75	0.65	0.81	0.76	0.82	0.71	0.82	0.71	0.80	0.68	0.84	0.45	0.89	0.83
C3	**0.94**	0.77	0.56	0.38	0.83	0.58	0.61	0.42	0.61	0.42	0.61	0.48	0.62	0.51	0.62	0.51	0.62	0.51	0.92	0.39	0.87	**0.79**
C4	**0.77**	0.58	0.63	0.41	0.73	0.38	0.65	0.28	0.65	0.28	0.32	0.17	0.22	0.14	0.38	0.23	0.38	0.23	**0.77**	**0.61**	0.70	0.54
Average	**0.81**	**0.71**	0.46	0.27	0.77	0.51	0.69	0.52	0.69	0.51	0.65	0.55	0.65	0.53	0.66	0.54	0.66	0.54	0.78	0.46	0.78	0.67
Average rank	**2.75**	**2**	10.3	9.83	4.42	6.75	6.25	6.33	6.67	6.87	7.33	6.33	6.58	5.67	6.08	5.25	6.58	5.25	4.42	7.67	3.58	3.08

We compared the proposed OSM classifier with 10 algorithms (or combinations of algorithms) including (1) conventional SVM, (2) undersampling SVM, (3) SMOTE+SVM, (4) SMOTE with different costs (SDC) (Akbani et al., 2004) that integrate the SMOTE and DEC methods, (5) boosting SVM (Wang & Japkowicz, 2010) that combines a boosting method with asymmetric misclassification costs, (6) FSVM-CIL with distance from the center of each class (FSVM-CIL$_{cen}$), (7) FSVM-CIL with distance from the actual hyperplane (FSVM-CIL$_{hyp}$), (8) entropy-based FSVM (Fan et al., 2017) that assigns entropy-based fuzzy membership (EFSVM), (9) EMatMHKS (Zhu & Wang, 2017) that combines the FSVM-CIL method and the matrix-pattern-oriented Ho-Kashyap learning machine with regularization learning, and (10) 1-NN.

As for undersampling, resampling was applied until the training data sets were balanced. The testing data sets retained their original imbalanced distributions. For the kernel, we used the radial basis kernel function. For the undersampling algorithms, we undersampled the training data until both classes were equal in number. For SMOTE, we oversampled the data instead of undersampling them. The parameter "under" was fixed to 100. The optimal parameter "over" was found in the range from 100 to 1000 until both classes were equal. The number of nearest neighbors, k, was set to 5. As for the SDC method, C^-/C^+ was set to the positive-to-negative class ratio of the data set according to the findings of Akbani et al. (2004). The maximum iteration for boosting SVM was set to 100. In the FSVM-CIL method, β in the exponential decay function $f(x_i)$ was determined by 10-fold cross-validation with various values ($\beta = 0.1, 0.2, \ldots, 0.9, 1$). As for the EFSVM method, we examined different numbers of nearest neighbors k ($k = 3, 5, 7, 9$) and found one that yielded the minimum 10-fold cross-validated error rate. The separated subset m was empirically determined by 10, and the fuzzy membership parameter was also determined empirically as 0.05. For EMatMHKS, we varied the values of the three parameters (i.e., bias, left, and right weight) from 0 to 1 (0.1, 0.2,..., 0.9, 1), respectively, and found the ones that yielded the minimum 10-fold cross-validated error rate. The number of nearest neighbors k was chosen in the same way as in the EFSVM.

In these experiments, the performance of each method was evaluated by using a 10-fold cross-validation technique. More precisely, the training data set was randomly partitioned into 10 subsamples of equal size. In each round, one subsample out of 10 was used for testing. The accuracy rates from 10 testing data sets were averaged. All the experiments were implemented using the R software and run on a computer with an i5-4590 CPU and 24 GB of RAM.

Table 2 shows the classification performance of each model under 12 scenarios. The boldface represents the highest classification performance of the GM and F1 for each corresponding scenario. The average GM and F1 values and their average ranks over all data sets are listed in the last two rows, respectively. The OSM classifier showed the best classification performance on six synthetic data sets in terms of the GM and F1, respectively. Furthermore, the OSM classifier ranked first place among the compared algorithms. In the slightly overlapping case (Scenario A), the proposed OSM classifier showed slightly better classification performance than the other methods. When the degree of overlap was increasing (Scenarios B and C), the proposed OSM classifier showed better performance than the alternatives. In particular, in the highly imbalanced and overlapping scenario cases such as C1, C2, and C3, the GM and F1 of the OSM classifier outperformed the other classifiers.

Table 3
Description of the real-world data sets.

Data sets	Number of features	Number of observations in the minority class	Number of observations in the majority class	Total number of observations	Imbalance ratio	Average overlap-sensitive cost in the majority class	Average overlap-sensitive cost in the minority class
Wisconsin	9	239	444	683	1.86	0.974	0.925
glass0	9	70	144	214	2.06	0.773	0.681
heberman	3	81	225	306	2.78	0.786	0.344
vehicle1	18	217	629	846	2.9	0.792	0.45
vehicle3	18	212	634	846	2.99	0.826	0.392
vehicle0	18	199	647	846	3.25	0.931	0.8
newthyroid1	5	35	180	215	5.14	0.983	0.724
glass6	9	29	185	214	6.38	0.972	0.78
ecoli046vs5	6	20	183	203	9.15	0.98	0.742
vowel0	13	90	898	988	9.98	0.992	0.75
glass016vs2	9	17	175	192	10.29	0.92	0.219
glass2	9	17	197	214	11.59	0.934	0.23
shuttle0vs4	9	123	1706	1829	13.87	1	0.988
glass4	9	13	201	214	15.46	0.972	0.455
pageblocks13vs4	10	28	444	472	15.86	0.969	0.6
glass016vs5	9	9	175	184	19.44	0.975	0.314
shuttle2vs4	9	6	123	129	20.5	1	0.331
shuttle6vs23	9	10	220	230	22	1	0.855
yeast1458vs7	8	30	663	693	22.1	0.956	0.099
glass5	9	9	205	214	22.78	0.979	0.308
yeast4	8	51	1433	1484	28.1	0.978	0.239
winequalityred4	11	53	1546	1599	29.17	0.969	0.036
yeast5	8	44	1440	1484	32.73	0.988	0.568
ecoli0137vs26	7	7	274	281	39.14	0.99	0.434
abalone17vs78910	8	58	2280	2338	39.31	0.989	0.151
abalone21vs8	8	14	567	581	40.5	0.993	0.19
yeast6	8	35	1449	1484	41.4	0.983	0.452
winequalitywhite3vs7	11	25	1457	1482	58.28	0.989	0.097
winequalityred3vs5	11	10	681	691	68.1	0.987	0.027

4.2. Experiments on real-world data sets

We used 29 real-world data sets from the KEEL data set repository (Alcalá-Fdez et al., 2011) to demonstrate the applicability of the proposed method. Table 3 lists these data sets in ascending order of imbalance ratio.

A five-fold cross-validation was performed to compare the performances between the proposed OSM classifier and 10 comparative methods in terms of F1 and GM. We selected the parameters for each method in the same manner as in the above experiments on synthetic data. Table 4 shows the experimental results; the boldface represents the highest classification accuracy of the GM and F1 for each data set. The average GM and F1 values and their average ranks over all data sets are listed in the last two rows of the table. The average ranks of the GM and F1 of the proposed OSM classifier were 3.59 and 3.72, respectively, indicating that the OSM classifier yielded a better performance than the other methods. In the data sets whose imbalanced ratio was less than 20, the proposed OSM classifier yielded comparable performance with the other classifiers, except for the data sets *glass016vs2* and *glass2*. In this case, the overall average overlap-sensitive cost value in the minority class was 0.58, indicating a small overlap. When the degrees of imbalance and overlapping were relatively small, most of the samples were well classified by the OSM decision boundary only. Thus, the OSM results showed similar classification performance to that of the other algorithms. The data sets *glass016vs2* and *glass2* involved a large overlap because their overlap-sensitive costs were 0.219 and 0.23, respectively. In these data sets, the proposed OSM classifier yielded better performance than the other classifiers. The proposed OSM classifier outperformed the other methods in terms of F1 in highly imbalanced data sets whose imbalanced ratio was larger than 20. This indicated that the OSM classifier could correctly detect the minority (positive) class, which is of particular interest to users. In this case, the average overlap-sensitive cost value in the minority class was 0.29, indicating a large overlap. The proposed OSM classifier correctly classified the observations in the soft-overlap region and then constructed the flexible decision boundary in the hard-overlap region. In particular, for the extremely imbalance and overlapping data sets such as *yeast1458vs7*, *winequality-white3vs7*, *winequality-red4*, and *winequality-red3vs5*, the proposed OSM classifier outperformed the other classifiers. Furthermore, the OSM classifier showed better classification performance than the 1-NN algorithm in the highly imbalanced data sets. We believe this demonstrates the effectiveness of the proposed method, which uses the 1-NN algorithm only for the hard-overlap region. It is worth mentioning that the F1 values of the traditional SVM were zero, especially for the highly imbalanced data sets. This indicates that the traditional SVM performed poorly because it classified all observations into the majority class in highly imbalanced data sets.

To obtain statistical significance for differences in ranks, we conducted a statistical comparison test using the Friedman test for the GM and F1 (Demšar, 2006). We evaluated the null hypothesis H_0 and the alternate hypothesis H_a to determine whether the average ranks of the compared classifiers over all data sets were significantly different:

H_0: No difference exists in the average ranks for classifiers over the data sets
H_a: A difference exists in the average ranks for classifiers over the data sets

Because our experiments were conducted using 29 data sets and 11 methods, the Friedman statistic followed the F-distribution with 10 and 280 degrees of freedom. The Friedman statistics of the GM and F1 were calculated as 7.1182 and 13.007, respectively. The p-value computed by $F(10, 280)$ was less than 0.01, indicating that the null hypothesis should be rejected. Thus, the average ranks of the GM and F1 were statistically different for the compared classifiers.

Table 4
Comparison of 11 classifiers for 29 real-world data sets in terms of the GM and F. The boldface represents the highest accuracy rates of the GM and F1 for each corresponding data set.

Data sets	OSM		Normal SVM		Under+SVM		SMOTE+SVM		SDC		Boosting SVM		FSVM-CIL$_{cen}$		FSVM-CIL$_{hyp}$		EFSVM		EMatMHKS		1-NN	
	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1	GM	F1
Wisconsin	**0.96**	0.94	0.95	**0.96**	0.95	0.94	**0.96**	0.95	0.95	0.93	0.95	0.94	**0.96**	0.93	0.95	0.93	0.95	0.93	0.95	0.94	**0.96**	0.95
glass0	0.82	0.76	0.74	**0.8**	0.70	0.73	0.80	0.73	0.80	0.74	0.78	0.75	0.64	0.56	0.79	0.73	0.79	0.73	0.73	0.70	**0.83**	0.77
heberman	0.38	0.25	0.21	0.36	0.46	**0.4**	**0.49**	0.36	0.38	0.25	0.37	0.32	0.37	0.29	0.34	0.19	0.38	0.23	0.38	0.35	0.47	0.3
vehicle1	**0.73**	0.60	0.53	**0.65**	0.71	0.57	0.69	0.57	0.67	0.55	0.62	0.57	0.59	0.42	0.61	0.49	0.65	0.52	0.61	0.54	0.59	0.44
vehicle3	**0.76**	0.63	0.56	**0.68**	0.73	0.58	0.68	0.55	0.69	0.57	0.64	0.6	0.40	0.33	0.65	0.53	0.67	0.55	0.56	0.53	0.65	0.51
vehicle0	0.95	0.90	0.91	0.94	0.85	0.92	0.95	0.94	0.94	0.92	0.93	0.92	0.77	0.63	0.92	0.91	**0.98**	**0.97**	0.88	0.83	0.91	0.85
newthyroid1	0.97	0.91	0.93	0.95	0.97	0.92	0.97	**0.97**	**0.98**	**0.97**	0.90	0.87	0.81	0.62	0.80	0.75	**0.98**	0.94	0.90	0.83	**0.98**	0.94
glass6	0.85	0.77	0.81	0.83	**0.89**	0.69	0.88	**0.86**	0.88	**0.86**	0.86	0.79	0.79	0.60	0.86	0.84	0.88	**0.86**	0.83	0.71	0.88	0.84
ecoli046vs5	**0.89**	0.76	0.85	0.78	0.82	0.59	0.85	0.79	0.85	0.79	0.87	0.74	0.76	0.44	0.85	0.76	0.88	**0.8**	0.84	0.64	0.84	0.78
vowel0	0.97	0.88	0.99	0.99	0.98	0.92	0.99	0.99	0.99	0.99	0.99	0.97	0.94	0.74	0.99	0.99	0.99	0.99	0.97	0.88	**1.00**	**1.00**
glass016vs2	**0.71**	**0.46**	0.15	0.21	0.45	0.23	0.31	0.23	0.31	0.23	0.30	0.20	0.19	0.13	0.21	0.15	0.21	0.15	0.30	0.19	0.35	0.26
glass2	**0.60**	**0.42**	0.16	0.24	0.42	0.30	0.35	0.23	0.35	0.23	0.38	0.24	0.20	0.11	0.25	0.19	0.28	0.22	0.36	0.22	0.34	0.23
shuttle0vs4	**1.00**	0.97	0.99	**1.00**	**1.00**	0.99	**1.00**	**1.00**	**1.00**	**1.00**	0.99	0.99	0.99	0.89	**1.00**	**1.00**	**1.00**	**1.00**	0.99	0.96	**1.00**	**1.00**
glass4	0.75	0.62	0.69	**0.66**	0.68	0.47	0.56	0.53	0.56	0.53	0.80	0.67	0.70	0.35	0.72	0.67	0.67	0.59	0.79	0.56	**0.80**	0.67
pageblocks13vs4	**0.98**	0.91	0.89	0.94	0.88	0.74	0.92	0.89	0.92	0.89	0.93	0.86	0.90	0.51	0.92	0.91	0.96	0.94	0.92	0.73	**0.98**	**0.98**
glass016vs5	0.78	0.63	0.79	**0.88**	0.80	0.62	0.79	0.72	0.79	0.72	0.79	0.70	0.78	0.22	0.68	0.6	0.74	0.66	0.82	0.57	**0.87**	0.69
shuttle2vs4	0.93	0.90	0.79	0.9	0.87	0.72	0.94	**0.93**	0.8	0.73	**0.97**	0.85	0.96	0.7	0.94	**0.93**	0.94	**0.93**	**0.97**	0.77	0.94	**0.93**
shuttle6vs23	**1**	**1**	0.14	0.13	0.75	0.46	0.92	0.66	0.94	0.93	0.94	0.89	0.94	0.93	0.97	0.76	0.97	0.97	0.96	0.83	0.94	0.93
yeast1458vs7	**0.54**	**0.24**	0.20	0.15	0.51	0.12	0.28	0.15	0.28	0.15	0.33	0.14	0.15	0.07	0.20	0.15	0.20	0.15	0.32	0.11	0.27	0.14
glass5	0.87	0.72	0.67	**0.79**	**0.92**	0.43	0.74	0.69	0.8	0.76	0.76	0.61	0.71	0.36	0.68	0.61	0.74	0.69	0.77	0.53	0.79	0.71
yeast4	0.54	0.26	0.43	0.25	**0.67**	0.20	0.40	0.24	0.40	0.24	0.43	0.17	0.35	0.07	0.11	0.08	0.40	0.24	0.51	0.17	0.57	**0.35**
winequalityred4	0.42	**0.20**	0.00	0.00	0.56	0.08	0.65	0.11	0.65	0.13	0.66	0.13	0.65	0.12	0.64	0.11	0.54	0.16	0.65	0.12	0.27	0.11
yeast5	**0.87**	0.67	0.74	0.66	0.85	0.45	0.78	**0.71**	0.78	**0.71**	0.70	0.50	0.76	0.24	0.42	0.38	0.70	0.63	0.82	0.45	0.82	0.68
ecoli0137vs26	0.72	0.38	0.53	0.37	0.68	0.15	0.53	0.37	0.53	0.37	0.45	0.22	0.61	0.13	0.14	0.13	0.53	0.37	0.61	0.22	**0.73**	**0.56**
abalone17vs78910	0.60	**0.30**	0.00	0.00	0.64	0.08	0.63	0.08	**0.84**	0.2	0.65	0.08	0.67	0.08	0.52	0.06	0.64	0.19	0.59	0.07	0.42	0.22
abalone21vs8	0.69	**0.75**	0.00	0.00	0.73	0.10	0.78	0.16	**0.87**	0.48	0.8	0.20	0.79	0.17	0.8	0.19	0.74	0.46	0.80	0.20	0.61	0.58
yeast6	0.73	**0.46**	0.60	0.43	**0.75**	0.23	0.57	0.37	0.57	0.37	0.58	0.28	0.52	0.10	0.28	0.19	0.57	0.40	0.66	0.25	0.7	**0.46**
winequalitywhite3vs7	0.47	**0.50**	0.00	0.00	0.67	0.13	0.62	0.10	0.60	0.09	**0.70**	0.13	0.67	0.14	**0.70**	0.10	0.57	0.32	**0.70**	0.12	0.20	0.16
winequalityred3vs5	**0.70**	**0.29**	0.00	0.00	0.69	0.07	0.65	0.09	0.5	0.08	0.66	0.12	0.59	0.12	0.39	0.05	0.65	0.21	0.55	0.09	0.00	0.00
Average	**0.76**	**0.62**	0.53	0.54	0.74	0.48	0.71	0.55	0.71	0.57	0.71	0.53	0.66	0.38	0.63	0.50	0.70	0.58	0.72	0.49	0.68	0.59
Average rank	**3.59**	**3.72**	8.41	4.38	4.69	7.07	4.45	4.00	4.41	4.21	4.79	5.52	7.69	9.41	6.76	7.03	5.24	4.14	5.31	7.86	4.52	3.86

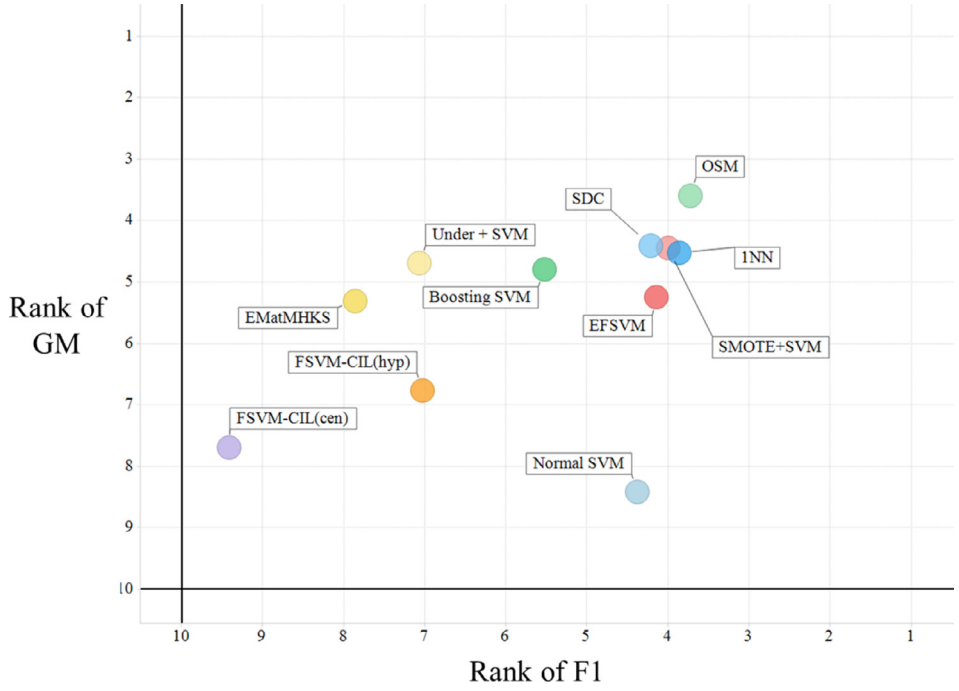
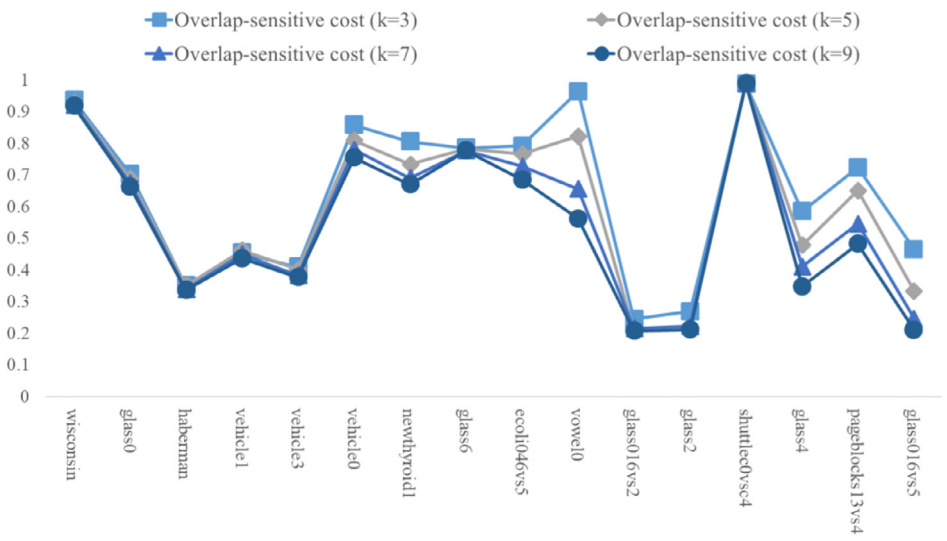


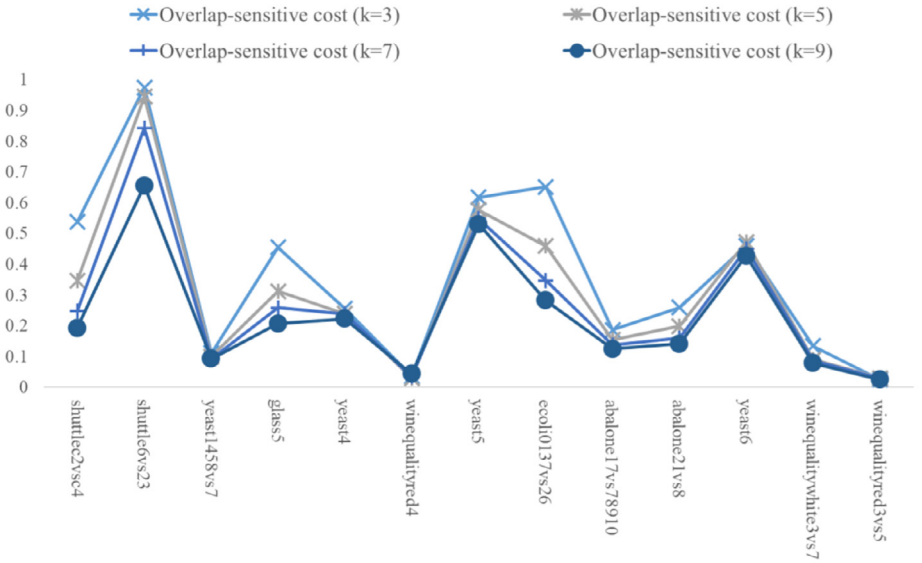
Fig. 8. Two-dimensional plot (average rank of F1 vs. average rank of GM) to facilitate the visualization of the comparative results.

Fig. 8 shows a two-dimensional plot (average rank of F1 vs. average rank of GM) to facilitate the interpretation of the comparative results, which indicate that the proposed OSM classifier performed better than the other classifiers because it is located in the upper right of the graph. It can be observed that the average rank of the GM in the proposed OSM significantly outperformed all the other classifiers. The proposed OSM yielded a slightly better average rank for F1, which exhibited a detection ability for the minority class (class of interest) compared with SDC, SMOTE+SVM, EFSVM, and 1-NN. Furthermore, it is worth noting that the proposed OSM outperformed the other methods, especially for the highly imbalanced data sets with an imbalanced ratio larger than 20. These included *shuttle6vs23, winequalityred4, abalone17vs78910, abalone21vs8, yeast6, winequalitywhite3vs7*, and *winequalityred3vs5*.

To construct the proposed OSM classifier, we used the kNN algorithm to measure the degree of overlap for each observation. To examine how much different values of k affected the overlap-sensitive costs, we calculated the average values of the



(a) The data sets whose imbalanced ratio was smaller than 20



(b) The data sets whose imbalanced ratio was larger than 20

Fig. 9. Average values of the overlap-sensitive cost in the minority class with different values of k for 29 real-world data sets: (a) the data sets whose imbalance ratio was smaller than 20 and (b) the data sets whose imbalance ratio was larger than 20.

Table 5
Comparison of the training time (in seconds) of 11 classifiers for 29 real-world data sets.

Method	Calculation time (seconds)	Rank
1-NN	0.04 ± 0.03	1
Under + SVM	0.54 ± 0.79	2
Normal SVM	1.18 ± 1.97	3
EmatMHKS	3.43 ± 4.66	4
OSM (proposed method)	3.82 ± 2.01	5
EFSVM	4.09 ± 3.26	6
FSVM-CIL$_{hyp}$	4.16 ± 6.17	7
FSVM-CIL$_{cen}$	4.55 ± 8.03	8
SMOTE+SVM	12.9 ± 24.78	9
SDC	13.65 ± 26.68	10
Boosting SVM	61.47 ± 92.11	11

overlap-sensitive cost in the minority class with different values of k for 29 real-world data sets (Fig. 9). To facilitate visualization, we generated two figures based on whether the imbalance ratio of the data was smaller than or larger than 20. Fig. 9 clearly shows that k did not significantly affect the average overlap-sensitive cost values.

Table 5 shows the average training times for each method for 29 real-world data sets. The training times for all the methods were comparable in that their training was completed within 62 s on average.

To examine the scalability of the proposed OSM classifier, we computed the empirical time complexity against the size of the training data sets. Fig. 10 shows that the OSM classifier was able to provide polynomial time complexity (i.e., a quadratic function). We could also roughly estimate the training time with larger sizes of data from this fitted quadratic function. For example, if the size of the training data set is 5000 and 10,000, the training time requires 24.5 s and 99.04 s, respectively. We thought that this is a reasonable complexity compared to other algorithms.

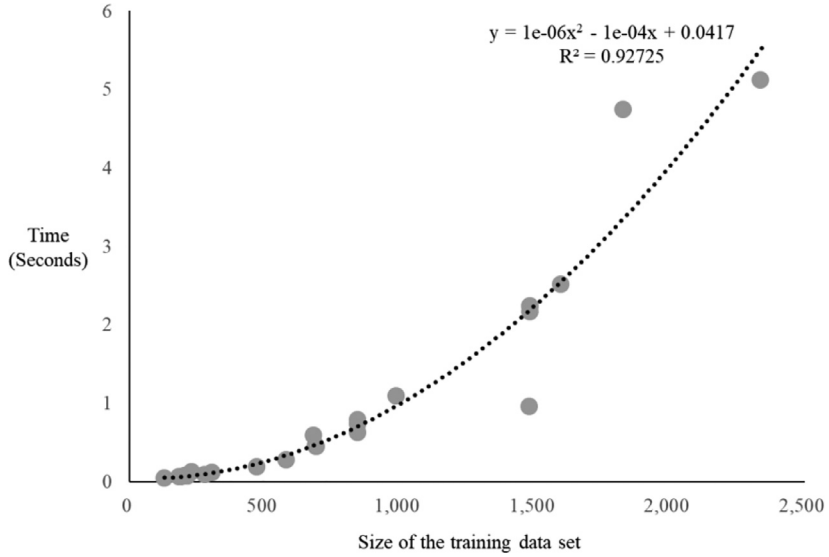


Fig. 10. Empirical time complexity of the proposed OSM classifier. The *x*-axis and *y*-axis represent the size of the training data set and the time complexity, respectively.

5. Conclusions

In this paper, we have proposed a method based on OSM margin to geometrically separate a data set containing imbalanced and overlapping data into soft- and hard-overlap regions. To verify the OSM classifier, we classified real-world data sets with various data distributions and degrees of overlap. The proposed OSM method outperformed other well-known machine learning SVM-based classifiers in terms of prediction accuracy (F1 and GM). Through empirical studies, we demonstrated that the proposed method is effective. The OSM classifier provides two major contributions: (1) most previous methods considered only the class imbalance problem; however, the main objective of the proposed method is to improve classification performance with data sets that have both class imbalance and class overlap; (2) by separating a data set into soft- and hard-overlap regions, we can indirectly decide whether or not to closely examine the observations. For example, in real-world applications, if the observations lie in the hard-overlap regions, the inspector should look more carefully at them.

Although the proposed method showed favorable classification results, it has some limitations. First, we applied the 1-NN algorithm for the hard-overlap region. Because 1-NN is an extremely local search algorithm, it can yield a high generalization error. In other words, if we can develop a more precise classification algorithm for highly overlapping regions, the classification results would be more stable. Second, we used the OSM margin to separate the given data sets. If we consider splitting a data set into nonparallel hyperplanes, e.g., by using a twin SVM, the quality of the proposed method could also be improved.

Acknowledgments

The authors would like to thank the editor and reviewers for their useful comments and suggestions, which were of great help in improving the quality of the paper. This work was supported by Brain Korea PLUS; by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT and Future Planning (NRF-2016R1A2B1008994); and by the Ministry of Trade, Industry & Energy under the Industrial Technology Innovation Program (R1623371).

References

Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying Support Vector Machine to Imbalanced Datasets. *Machine Learning: ECML*, 39–50.

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., et al. (2011). KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing, 17*(2–3), 255–287.

Ashfaq, R. A. R., Wang, X. Z., Huang, J. Z., Abbas, H., & He, Y. L. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences, 378*, 484–497.

Bae, M. H., Wu, T., & Pan, R. (2010). Mix-ratio sampling: Classifying multiclass imbalanced mouse brain images using support vector machine. *Expert Systems with Applications, 37*(7), 4955–4965.

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter - Special Issue on Learning from Imbalanced Datasets, 6*(1), 20–29.

Batuwita, R., & Palade, V. (2010). FSVM-CIL: Fuzzy support vector machines for class imbalance learning. *IEEE Transactions on Fuzzy Systems, 18*(3), 558–571.

Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications, 36*(3), 4626–4636.

Cao, P., Zhao, D., & Zaiane, O. (2013). An optimized cost-sensitive SVM for imbalanced data learning. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in Bioinformatics): Vol. 7819* (pp. 280–292). LNAI.

Chan, P. K., Fan, W., Prodromidis, A. L., & Stolfo, S. J. (1999). Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems, 14*(6), 67–74.

Chand, N., Mishra, P., Krishna, C. R., Pilli, E. S., & Govil, M. C. (2016). A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection. In *Proceedings - 2016 international conference on advances in computing, communication and automation, ICACCA 2016*.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

Chawla, N. V., Japkowicz, N., & Drive, P. (2004). Editorial : Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter, 6*(1), 1–6.

Dai, H.-L. (2015). Class imbalance learning via a fuzzy total margin based support vector machine. *Applied Soft Computing, 31*, 172–184.

Dangare, C. S., & Apte, S. S. (2012). Improved study of heart disease prediction system using data mining classification techniques. *International Journal of Computer Applications (0975 - 888), 47*(10), 44–48.

Das, B., Krishnan, N. C., & Cook, D. J. (2013). Handling class overlap and imbalance to detect prompt situations in smart homes. In *Proceedings - IEEE 13th international conference on data mining workshops, ICDMW 2013* (pp. 266–273).

Datta, S., & Das, S. (2015). Near-Bayesian support vector machines for imbalanced data classification with equal or unequal misclassification costs. *Neural Networks, 70*, 39–52.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

Deng, X., Tian, X., Chen, S., & Harris, C. J. (2016). Statistics local fisher discriminant analysis for industrial process fault classification. In Control *(CONTROL)*, 2016 UKACC 11th international conference on (pp. 1–6). IEEE.

Dorronsoro, J. R., Ginel, F., Sánchez, C., & Santa Cruz, C. (1997). Neural fraud detection in credit card operations. *IEEE Transactions on Neural Networks, 8*(4), 827–834.
Elkan, C. (2001). The foundations of cost-sensitive learning. In *IJCAI international joint conference on artificial intelligence* (pp. 973–978).
Fan, Q., Wang, Z., Li, D., Gao, D., & Zha, H. (2017). Entropy-based fuzzy support vector machine for imbalanced datasets. *Knowledge-Based Systems, 115*, 87–99.
García, V., Mollineda, R. A., & Sánchez, J. S. (2008). On the k-NN performance in a challenging scenario of imbalance and overlapping. *Pattern Analysis and Applications, 11*(3–4), 269–280.
Gür Ali, Ö., & Aritürk, U. (2014). Dynamic churn prediction framework with more effective use of rare event data: The case of private banking. *Expert Systems with Applications, 41*(17), 7889–7903.
Guo, X., Yin, Y., Dong, C., Yang, G., & Zhou, G. (2008). On the class imbalance problem. In *Proceedings - 4th international conference on natural computation, ICNC 2008: Vol. 4* (pp. 192–201).
He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263–1284.
Imam, T., Ting, K. M., & Kamruzzaman, J. (2006). Z-SVM: An SVM for improved classification of imbalanced data. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics): Vol. 4304* (pp. 264–273). LNAI.
Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis, 6*(5), 429–449.
Kang, P., & Cho, S. (2006). EUS SVMs : Ensemble of Under-Sampled SVMs for Data Imbalance Problems. In *Proceedings of the international conference on neural information processing (ICONIP): 298* (pp. 837–846).
Koh, H. C., & Tan, G. (2005). Data mining applications in healthcare. *J Healthc Inf Manag, 19*(2), 64–72.
Kukar, M., & Kononenko, I. (1998). Cost-sensitive learning with neural networks. In *13th European conference on artificial intelligence* (pp. 445–449).
Lin, C. F., & Wang, S.-D. (2002). Fuzzy support vector machines. *Neural Networks, IEEE Transactions on, 13*(2), 464–471.
Liu, Y., Yu, X., Huang, J. X., & An, A. (2011). Combining integrated sampling with SVM ensembles for learning from imbalanced datasets. *Information Processing and Management, 47*(4), 617–631.
Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A., & Tourassi, G. D. (2008). Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks, 21*(2–3), 427–436.
Napierała, K., & Stefanowski, J. (2015). Addressing imbalanced data with argument based rule learning. *Expert Systems with Applications, 42*(24), 9468–9481.
Nekooeimehr, I., & Lai Yuen, S. K. (2016). Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets. *Expert Systems with Applications, 46*, 405–416.
Osuna, E., Freund, R., & Girosit, F. (1997). Training support vector machines: An application to face detection. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition* (pp. 130–136).
Qu, Y., Su, H., Guo, L., & Chu, J. (2011). A novel SVM modeling approach for highly imbalanced and overlapping classification. *Intelligent Data Analysis, 15*(3), 319–341.
Rivera, W. A., & Xanthopoulos, P. (2016). A priori synthetic over-sampling methods for increasing classification sensitivity in imbalanced data sets. *Expert Systems with Applications, 66*, 124–135.
Rostami, H., Blue, J., & Yugma, C. (2016). Equipment condition diagnosis and fault fingerprint extraction in semiconductor manufacturing, 534–539.
Sahin, Y., Bulkan, S., & Duman, E. (2013). A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications, 40*(15), 5916–5923.
Sun, Y., Kamel, M. S., Wong, A. K. C., & Wang, Y. (2007). Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition, 40*(12), 3358–3378.
Tang, W., & Mao, K. (2010). Classification for overlapping classes using optimized overlapping region detection and soft decision. *Fusion (FUSION) 2010*.
Tang, Y., Zhang, Y. Q., & Chawla, N. V. (2009). SVMs modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 39*(1), 281–288.
Tomek, I. (1976). Two modifications of CNN. *IEEE Transactions on Systems, Man and Cybernetics, 6*, 769–772.
Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 45–66.
Turney, P. (1995). Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research, 2*, 369–409.
Van, M., & Kang, H. (2016). Bearing Defect classification based on individual wavelet local fisher discriminant analysis with particle swarm optimization, 12(1), 124–135.
Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer.
Veropoulos, K., Cristianini, N., & Campbell, C. (1999). Controlling the sensitivity of support vector machines. In *Proceedings of the international joint conference on artificial intelligence* (pp. 55–60).
Wang, B. X., & Japkowicz, N. (2010). Boosting support vector machines for imbalanced data sets. *Knowledge and Information Systems, 25*(1), 1–20.
Wang, H., Gao, Y., Shi, Y., & Wang, H. (2017). A fast distributed classification algorithm for large-scale imbalanced data. In *Proceedings - IEEE international conference on data mining, ICDM* (pp. 1251–1256).
Wilinski, A., & Osowski, S. (2009). Gene selection for cancer classification. *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering, 28*(1), 231–241.
Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics, 2*(3), 408–421.
Xiong, H., Wu, J., & Liu, L. (2010). Classification with class overlapping: A systematic study. In *The 2010 international conference on e-business intelligence* (pp. 491–497).
Yang, X., Han, L., Li, Y., He, L., Yang, X., Han, L., et al. (2015). A bilateral - truncated - loss based robust support vector machine for classification problems. *Soft Computing*, 2871–2882.
Zhou, Z. H., & Liu, X. Y. (2006). Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering, 18*(1), 63–77.
Zhu, C., & Wang, Z. (2017). Entropy-based matrix learning machine for imbalanced data sets. *Pattern Recognition Letters, 88*, 72–80.