Journal of Machine Learning Research 16 (2015) 2695-2724 Submitted 4/13; Revised 12/14; Published 12/15

Absent Data Generating Classifier for Imbalanced Class Sizes

Arash Pourhabib ARASH.POURHABIB@OKSTATE.EDU
School of Industrial Engineering and Management
Oklahoma State University
322 Engineering North, Stillwater, Oklahoma 74078-5016, USA

Bani K. Mallick BMALLICK@STAT.TAMU.EDU
Department of Statistics
Texas A&M University
3143 TAMU, College Station, TX 77843-3143, USA

Yu Ding YUDING@IEMAIL.TAMU.EDU
Department of Industrial and Systems Engineering
Texas A&M University
3131 TAMU, College Station, TX 77843-3131, USA

Editor: Russ Greiner

Abstract

We propose an algorithm for two-class classification problems when the training data are imbalanced. This means the number of training instances in one of the classes is so low that the conventional classification algorithms become ineffective in detecting the minority class. We present a modification of the kernel Fisher discriminant analysis such that the imbalanced nature of the problem is explicitly addressed in the new algorithm formulation. The new algorithm exploits the properties of the existing minority points to learn the effects of other minority data points, had they actually existed. The algorithm proceeds iteratively by employing the learned properties and conditional sampling in such a way that it generates sufficient artificial data points for the minority set, thus enhancing the detection probability of the minority class. Implementing the proposed method on a number of simulated and real data sets, we show that our proposed method performs competitively compared to a set of alternative state-of-the-art imbalanced classification algorithms.

Keywords: kernel Fisher discriminant analysis, imbalanced data, two-class classification

1. Introduction

Classification is a task of supervised learning in which the response function assumes a set of integer values known as the class labels. In particular, two-class classification refers to algorithms producing binary responses and aiming at separating two probability densities after observing some instances from each class. In this paper, we are interested in developing a classification algorithm for a two-class classification problem in which the number of data points in one class (i.e. the majority class) is greater than those of the other class (i.e. the minority class). This type of data structure is called imbalanced data.

It is particularly crucial to correctly identify test cases belonging to the minority class as a low detection rate for the minority class could incur heavy expenses in practice. The reason lies in the nature of the minority classes. For example, in quality control applications, the minority class is the class of defective products; in security applications, the minority class is the class of potential perpetrators or attackers; in medical applications, the minority class is the class of diseases or cancerous cells. A classification method that fails to detect the minority classes is useless for practical purposes.

If one is interested in detecting minority cases in application, a direct use of traditional two-class classifications, such as support-vector machines or logistic regression, is not reliable because when the minority class data are too few in the training set, those methods tend to label almost all the instances in the test set, minority or otherwise, as the majority class (Chen et al., 2005). A training data set overwhelmed with one class of data points and deficient in the other class misleads the two classification algorithms about the accurate boundary between the two groups. Using most standard loss functions, these classification algorithms see little penalty by classifying regions in which both the minority and majority points have high density.

The major efforts aimed at solving the imbalanced classification problem can be categorized into: (a) cost-sensitive methods and (b) sampling strategies (He and Garcia, 2009; Japkowicz, 2000). Cost-sensitive methods take the imbalance structure into account by assigning a higher cost to the miss-classification of minority data points (Elkan, 2001; Ting, 2002). Despite a theoretical connection between imbalanced structure and cost-sensitive framework (Maloof, 2003; Weiss, 2004), this class of algorithms however may fail in practice; for example, if in the training stage the instances forming the classes are separable (Wallace et al., 2011, p.757). More critically, determining a suitable cost function is not a straightforward task and it may be difficult to achieve a robust algorithm using cost-sensitive methods.

The basic idea of the sampling-based approach is to alter the imbalanced structure of the problem by using different types of sampling methods. Hence, the algorithms in this category can be classified according to the specific sampling approaches, including resampling with replication, undersampling, or synthetic oversampling. In resampling with replication, one can use, for instance, bootstrapping for oversampling the minority data (Chen et al., 2005; Byon et al., 2010). In undersampling, one downsamples the majority data points to create more balanced data sets and alleviate the imbalance attached to the original data (Liu et al., 2009).

A novel approach proposed by Chawla et al. (2002) and called SMOTE, generates extra synthetic minority data points by interpolating the spaces between existing minority data points. Unlike other sampling methods which resample the existing data, SMOTE “creates” new data points, debuting the synthetic oversampling approach. Since SMOTE, many other variations of synthetic oversampling have been proposed in the literature; among others, Han et al. (2005) proposed an algorithm generating minority data points close to the boundary of the two classes and Batista et al. (2004) utilized different heuristics to integrate with synthetic oversampling.

SMOTE has proven to be a powerful method for handling imbalanced classification problems and still serves as a benchmark for this class of problems. An important revelation from the success of SMOTE and the like is that the synthetic oversampling is more potent than merely resampling existing data. The power of synthetic oversampling seems to lie in the simple fact that extra data are synthesized. From another perspective, synthetic data generation can be considered as a case of “phantom-transduction” as opposed to the inductive inference (Akbani et al., 2004). In other words, generating extra synthetic minority data points resembles that of using test sets in learning (Gammerman et al., 1998). SMOTE, for instance, does not employ a sophisticated approach for data synthesizing, but uses a simple, yet proved highly effective in practice, data interpolation (Chawla et al., 2002). It is not clear, however, whether the mechanism of data synthesizing matters and if so, which type of mechanism to use.

The current literature does not seem to present a consensus concerning the effectiveness of data synthesizing mechanisms. We tend to believe that it matters, because if a data synthesizing mechanism is tailored to and/or embedded in a specific classification problem, we expect to observe improvements in classification performance. Some empirical evidence supports our belief (Han et al., 2005). At a minimum, we believe that the data synthesizing issue remains unsettled and is worth further investigation.

We also believe that an important question to ponder is how to decide the decision boundary if we were furnished with more instances of the minority class. It should be emphasized, however, that not all those could-be minority points carry the same amount of information; those that can guide the algorithm to expand the minority class's region are more valuable because it is the difficulty that classification algorithms confront. Basically the question becomes how to use the current data points to synthesize the "valuable" but absent minority data points that allow us to obtain a tighter boundary for the majority class.

Towards that goal, we employ the kernel trick embedded in Fisher discriminant analysis (Hofmann et al., 2008; Mika et al., 1999) in our data synthesizing mechanism in order to exploit the properties of newly generated points in the feature space without actually specifying them. We utilize two properties of the "artificially" generated minorities: (i) the points should be located as close as possible to the boundary of the majority class, (ii) their projection onto a lower dimensional space should be close to that of the existing minority points in their vicinity. Then we sample more minority points from the augmented data set, conditional on the boundary achieved. We perform this procedure iteratively until the algorithm achieves the desired performance, and label the resulting algorithm Absent Data Generator (ADG).

The remainder of this paper is organized as follows. Section 2 outlines the kernel Fisher discriminant analysis, formally defines the imbalanced classification problem and presents the main optimization formulation. Section 3 presents the details of the proposed algorithm. Section 4 describes the proposed method's application to several simulated and real data sets and the results when the data structure is imbalanced. Section 5 discusses finding a bound on the generalization error of the algorithm. Section 6 concludes the paper and offers suggestions for future research.

2. Problem Formulation

Let $\mathcal{X}$ denote the input space, and suppose $\mathcal{X}^- = \{\boldsymbol{x}_1^-, \boldsymbol{x}_2^-, \ldots, \boldsymbol{x}_{l_-}^-\} \subset \mathcal{X}$ is the training set of majority data points which are independent and identically distributed (i.i.d.) (negative points, labeled as -1 or simply "$-$") and $\mathcal{X}^+ = \{\boldsymbol{x}_1^+, \boldsymbol{x}_2^+, \ldots, \boldsymbol{x}_{l_+}^+\} \subset \mathcal{X}$ is the training set of minority data points, also i.i.d. (positive points, labeled as $+1$, or simply "$+$"). For notation simplicity, the subscript "$+$" on l_+ is dropped when the context is clear. In the case of imbalance data, we have $l_+ \ll l_-$, or simply $l \ll l_-$. The goal in this section is to introduce a basic framework and general thoughts on how to generate and then utilize artificial data points. We propose generating artificial data points near the discriminative boundary of the two classes, and that they are generated within existing clusters with the probability of artificial data generated within a cluster inversely proportional to the size of that cluster.

First, we introduce the notion of "absent data": intuitively, absent data refer to the data points belonging to the minority class whose lack of presence has made the problem imbalanced, and we intend to re-generate them for the purpose of two-class classification. The concept of some data being absent is based on the thought that the existing data points may convey some information that allows us to identify some new data points belonging to the same class. Of course, acknowledging the existing of absent data does not imply that we know their numbers or exact locations in the space *a priori*. But in the context of imbalanced classification, this assumption paves the way for solving the problem through synthetic oversampling of minority data. Let $\mathcal{Z} = \{\boldsymbol{x}^+_{l+1}, \boldsymbol{x}^+_{l+2}, \ldots, \boldsymbol{x}^+_{l+k}\} \subset \mathcal{X}^+$ denote these absent data from the minority class; we assume the absent data are also an i.i.d. sample. We may denote each $\boldsymbol{x}^+_{l+j} \in \mathcal{Z}$ by $\boldsymbol{z}_j$ for $j = 1, 2, \ldots, k$.

We first review the Fisher discriminant analysis briefly. For a two-class classification, Fisher linear discriminant can be expressed simply through the following optimization problem:

$$\max_{\boldsymbol{w}} J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}}, \tag{1}$$

where $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ are the between and within class scatter matrices, respectively:

$$\boldsymbol{S}_B = (\boldsymbol{m}_- - \boldsymbol{m}_+)(\boldsymbol{m}_- - \boldsymbol{m}_+)^T,$$

$$\boldsymbol{S}_W = \sum_{i \in \{-,+\}} \sum_{\boldsymbol{x} \in \mathcal{X}^i} (\boldsymbol{x} - \boldsymbol{m}_i)(\boldsymbol{x} - \boldsymbol{m}_i)^T, \tag{2}$$

and $\boldsymbol{m}_i = \frac{1}{l_i} \sum_{j=1}^{l_i} \boldsymbol{x}^i_j$, for $i \in \{-,+\}$, is the sample average of each class. Problem (1) can be interpreted as maximizing the ratio of the between-class variance to the pooled variance about the means. Under certain conditions, we can also interpret this formulation as an optimal Bayes classifier (Bickel and Levina, 2004). We will revisit this formulation in Section 5 when developing an error bound.

To deal with nonlinear cases, one can map the data into a high-dimensional feature space and perform the calculation in that space. However, if an appropriate kernel is chosen for the transformation of the data and the calculation only requires kernel evaluations, we do not have to perform any calculations in the high-dimensional feature space (Hofmann et al., 2008). This property, known as the kernel trick, can be applied to the Fisher discriminant analysis, resulting in the Kernel Fisher Discriminant (KFD) (Mika et al., 1999). Specifically, the KFD is the extension of the Fisher linear discriminant performed in the feature space which solves

$$\max_{\boldsymbol{w}} J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}^{\phi}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}^{\phi}_W \boldsymbol{w}}, \tag{3}$$

where $\boldsymbol{S}_B^{\boldsymbol{\phi}}$ and $\boldsymbol{S}_W^{\boldsymbol{\phi}}$ are the between and within class scatter matrices, respectively, in the feature space:

$$\boldsymbol{S}_B^{\boldsymbol{\phi}} = (\boldsymbol{m}_-^{\boldsymbol{\phi}} - \boldsymbol{m}_+^{\boldsymbol{\phi}})(\boldsymbol{m}_-^{\boldsymbol{\phi}} - \boldsymbol{m}_+^{\boldsymbol{\phi}})^T,$$

$$\boldsymbol{S}_W^{\boldsymbol{\phi}} = \sum_{i \in \{-,+\}} \sum_{\boldsymbol{x} \in \mathcal{X}^i} (\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{m}_i^{\boldsymbol{\phi}})(\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{m}_i^{\boldsymbol{\phi}})^T, \tag{4}$$

and $\boldsymbol{m}_i^{\phi} = \frac{1}{l_i} \sum_{j=1}^{l_i} \boldsymbol{\phi}(\boldsymbol{x}_j^i)$. Here, $\boldsymbol{\phi}$ is a nonlinear mapping from $\mathcal{X}$ to the feature space $\mathcal{F}$, which is assumed to be a separable Hilbert space endowed with an inner product $\langle \cdot, \cdot \rangle$ such that there exists a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ where $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}') \rangle$. Obviously, in this case $\boldsymbol{w} \in \mathcal{F}$. Applying to imbalanced data sets, KFD suffers the same problem as most other classifiers do, i.e. it falls short of detecting most minority points in the test stage.

Our goal is to consider the imbalanced structure explicitly and extend KFD in such a way that it could be applied to imbalanced data. Towards this end, our thought process is as follows: first, if we had extra data points from the minority class, those points would be projected with high probability to where the existing minority points are projected; second, points close to the boundary of the majority points carry more "information" so we can use them to find the separating hyperplane in the feature space. The latter is in fact an intuitive property we are seeking, but the former requires more clarification. Particularly, if dealing with complex patterns in high dimensions, we may frequently observe that the minority data points constitute separate clusters after (or before) being projected to a lower-dimensional space. Therefore, if resemblance in projection regions is used as a property to generate artificial data, it entails precaution against the effect of complex structures. One way to address this issue is to take the cluster-based structure of the data into account explicitly.

Suppose the training minority points constitute C different clusters, for $C \geq 1$. That is, we have $\mathcal{X}^+ = \bigcup_{c=1}^C \mathcal{X}_c^+$ and $\mathcal{X}_{c'}^+ \cap \mathcal{X}_c^+ = \emptyset$ for $c \neq c'$, where $\mathcal{X}_c^+ = \{\boldsymbol{x}_{1,c}^+, \boldsymbol{x}_{2,c}^+, \cdots, \boldsymbol{x}_{l_c,c}^+\}$ is the c-th cluster of the minority data points, and we have $|\mathcal{X}_c^+| = l_c$, and $\sum_{c=1}^C l_c = l$. Accordingly, we partition the absent data in $\mathcal{Z}$ also into C different clusters, $\mathcal{Z}_c$'s, for $c = 1, 2, \ldots, C$, each of which corresponds to one of the C clusters of the minority points. Specifically $\mathcal{Z}_c = \{\boldsymbol{x}_{l_c+1,c}^+, \boldsymbol{x}_{l_c+2,c}^+, \ldots, \boldsymbol{x}_{l_c+k_c,c}^+\}$, $\bigcup_{c=1}^C \mathcal{Z}_c = \mathcal{Z}$, and $\mathcal{Z}_{c'} \cap \mathcal{Z}_c = \emptyset$, for $c \neq c'$. We also have $|\mathcal{Z}_c| = k_c$, and $\sum_{c=1}^C k_c = k$. The previously defined notation $\boldsymbol{z}_j$ can be similarly extended as $\boldsymbol{z}_{j,c} := \boldsymbol{x}_{l_c+j,c}^+$, for $j = 1, 2, \cdots, k_c$.

To enforce the property that newly generated points would be projected with high probability to where the existing minority points are projected, we add the constraint

$$\left(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{z}_{j,c}) - \boldsymbol{w}^T \boldsymbol{m}_{+,c}^{\boldsymbol{\phi}}\right)^2 \leq \delta, \quad \text{for} \quad j = 1, 2, \ldots k_c, \quad c = 1, 2, \ldots C, \tag{5}$$

for some positive $\delta > 0$, where $\boldsymbol{m}_{+,c}^{\boldsymbol{\phi}} = \frac{1}{l_c} \sum_{j=1}^{l_c} \boldsymbol{\phi}(\boldsymbol{x}_{j,c}^+)$, namely the mean of cluster c in the feature space. To have the second property, i.e. to have more points close to the boundary of the majority points, we add another constraint,

$$(\boldsymbol{\phi}(\boldsymbol{z}_{j,c}) - \boldsymbol{m}_-^{\boldsymbol{\phi}})^T (\boldsymbol{\phi}(\boldsymbol{z}_{j,c}) - \boldsymbol{m}_-^{\boldsymbol{\phi}}) \leq \Lambda \quad \text{for} \quad j = 1, 2, \ldots k_c, \quad c = 1, 2, \ldots C, \tag{6}$$

for some positive $\Lambda > 0$. Constraint (5) ensures that the point $\boldsymbol{\phi}(\boldsymbol{z}_{j,c})$ is at most δ distance away from the current cluster center of a minority group. This constraint also incorporates the cases where the minority data are cohesive and do not constitute many clusters, i.e.

only one cluster is determined according to the algorithm discussed in Section 3, which means that the constraint implies that the newly generated data point is at most δ distance away from the mean of the minority data points. Constraint (6) ensures that the newly generated points are "useful" in the sense that they are located close to the boundary of the two groups.

As a result of the Representer's Theorem (Hofmann et al., 2008), we can safely assume both $\boldsymbol{w}$ and $\boldsymbol{\phi}(\boldsymbol{z}_{j,c})$ belong to the space generated by the training points, namely $\mathcal{X}^- \cup \mathcal{X}^+$, whose elements, with a slight abuse of notation, can be represented by $\{\boldsymbol{x}_p\}_{p=1}^n$ where $n = l_- + l$. Specifically,

$$\boldsymbol{w} = \sum_{p=1}^{n} \alpha_p \boldsymbol{\phi}(\boldsymbol{x}_p), \tag{7}$$

and

$$\boldsymbol{\phi}(\boldsymbol{z}_{j,c}) - \boldsymbol{m}_-^{\phi} = \sum_{p=1}^{n} \beta_p^{j,c} \boldsymbol{\phi}(\boldsymbol{x}_p), \quad \text{for} \quad j = 1, 2, \dots k_c, \quad c = 1, 2, \dots C, \tag{8}$$

where α_p and $\beta_p^{j,c}$ are real coefficients for $p = 1, 2, \dots, n$, $j = 1, 2, \dots, k_c$ and $c = 1, 2, \dots C$. Having made these assumptions, we can express constraints (5) and (6) as

$$\left[\sum_{p=1}^{n} \alpha_p \boldsymbol{\phi}(\boldsymbol{x}_p)^T \left(\sum_{p=1}^{n} \beta_p^{j,c} \boldsymbol{\phi}(\boldsymbol{x}_p) + \frac{1}{l_-} \sum_{\ell=1}^{l_-} \boldsymbol{\phi}(\boldsymbol{x}_\ell^-) - \frac{1}{l_c} \sum_{\ell=1}^{l_c} \boldsymbol{\phi}(\boldsymbol{x}_\ell^+) \right) \right]^2 \leq \delta, \tag{9}$$

and

$$\sum_{p=1}^{n} (\beta_p^{j,c})^2 K(\boldsymbol{x}_p, \boldsymbol{x}_p) \leq \Lambda, \quad \text{for} \quad j = 1, 2, \dots k_c, \quad c = 1, 2, \dots C, \tag{10}$$

respectively. In the matrix forms, the above two expressions can be represented as

$$\left[\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\beta}^{j,c} + \boldsymbol{\alpha}^T (\boldsymbol{M}_- - \boldsymbol{M}_+^c) \right]^2 \leq \delta, \quad \text{for} \quad j = 1, 2, \dots k_c, \quad c = 1, 2, \dots C, \tag{11}$$

and

$$(\boldsymbol{\beta}^{j,c})^T \boldsymbol{K} (\boldsymbol{\beta}^{j,c}) \leq \Lambda, \quad \text{for} \quad j = 1, 2, \dots k_c, \quad c = 1, 2, \dots C, \tag{12}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_n]^T$ and $\boldsymbol{\beta}^{j,c} = [\beta_1^{j,c}, \beta_2^{j,c}, \dots \beta_n^{j,c}]^T$, and $\boldsymbol{M}_-$ is an $n \times 1$ vector such that $(\boldsymbol{M}_-)_j = \frac{1}{l_-} \sum_{\ell=1}^{l_-} K(\boldsymbol{x}_j, \boldsymbol{x}_\ell^-)$, and $\boldsymbol{M}_+^c$ is an $n \times 1$ vector such that $(\boldsymbol{M}_+^c)_j = \frac{1}{l_c} \sum_{\ell=1}^{l_c} K(\boldsymbol{x}_j, \boldsymbol{x}_{\ell,c}^+)$. The $n \times n$ matrix $\boldsymbol{K}$ consists of all of the pairwise kernel evaluations, namely $(\boldsymbol{K})_{r,s} = K(\boldsymbol{x}_r, \boldsymbol{x}_s)$, for $r, s \in \{1, 2, \dots, n\}$.

Following the notation introduced in Mika et al. (1999),

$$\boldsymbol{M} := (\boldsymbol{M}_- - \boldsymbol{M}_+)(\boldsymbol{M}_- - \boldsymbol{M}_+)^T, \text{ and} \tag{13}$$

$$\boldsymbol{N} := \sum_{i \in \{-,+\}} \boldsymbol{K}_i (\boldsymbol{I} - \boldsymbol{1}_{l_i}) \boldsymbol{K}_i^T, \tag{14}$$

where $\boldsymbol{M}_+$ is an $n \times 1$ vector such that $(\boldsymbol{M}_+)_j = \frac{1}{l} \sum_{\ell=1}^{l} K(\boldsymbol{x}_j, \boldsymbol{x}_\ell^+)$, $\boldsymbol{K}_i$ is an $n \times l_i$ matrix with $(\boldsymbol{K}_i)_{r,s} = K(\boldsymbol{x}_r, \boldsymbol{x}_s^i)$ for $r \in \{1, 2, \dots, n\}$, $s \in \{1, 2, \dots, l_i\}$ for $i \in \{-, +\}$, $\boldsymbol{I}$ is the identity matrix of appropriate size, and $\mathbf{1}_{l_i}$ is a matrix of appropriate size whose entries are $\frac{1}{l_i}$ for $i = -$ and $i = +$, respectively. Now, we can formulate the classification problem with imbalanced data through the following optimization

$$\max_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^T \boldsymbol{M} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \boldsymbol{N} \boldsymbol{\alpha}}, \tag{15}$$

subject to

$$\left[\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\beta}^{j,c} + \boldsymbol{\alpha}^T (\boldsymbol{M}_- - \boldsymbol{M}_+^c)\right]^2 \leq \delta, \quad \text{for} \quad j = 1, 2, \ldots k_c, \quad c = 1, 2, \ldots C, \tag{16}$$

$$(\boldsymbol{\beta}^{j,c})^T \boldsymbol{K} (\boldsymbol{\beta}^{j,c}) \leq \Lambda, \quad \text{for} \quad j = 1, 2, \ldots k_c, \quad c = 1, 2, \ldots C. \tag{17}$$

To solve optimization problem (15)-(17), we assume $\delta = 0$. This implies that the newly generated points $\boldsymbol{z}_{j,c}$ should be projected where the mean of the corresponding cluster in the minority group is projected. As such, constraint (16) is replaced by

$$\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\beta}^{j,c} + \boldsymbol{\alpha}^T (\boldsymbol{M}_- - \boldsymbol{M}_+) = 0, \quad \text{for} \quad j = 1, 2, \ldots k_c, \quad c = 1, 2, \ldots C.$$

This new constraint is not restricting, since we next solve a relaxation of the original problem. Specifically, we use the Lagrangian relaxation (Anstreicher and Wolkowicz, 1998) for solving ((15))-(17). First, note that an equivalent way of writing the optimization ((15))-(17) is to consider the denominator in the objective function (15) as another constraint and only have the numerator in the objective function. Specifically, we consider the objective function to be

$$\max_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T \boldsymbol{M} \boldsymbol{\alpha}, \tag{18}$$

and add the constraint

$$\boldsymbol{\alpha}^T \boldsymbol{N} \boldsymbol{\alpha} \leq R, \tag{19}$$

to the optimization problem (15), for some positive number R. Having done that, we get the following for the Lagrangian function

$$\begin{aligned} J(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{\alpha}^T \boldsymbol{M} \boldsymbol{\alpha} \quad &- \quad \gamma \left[\boldsymbol{\alpha}^T \boldsymbol{N} \boldsymbol{\alpha} - R\right] \\ &- \quad \sum_{c=1}^{C} \sum_{j=1}^{k_c} \lambda_j^c \left[\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\beta}^{j,c} + \boldsymbol{\alpha}^T (\boldsymbol{M}_- - \boldsymbol{M}_+^c)\right] \\ &- \quad \sum_{c=1}^{C} \sum_{j=1}^{k_c} \mu_j^c \left[(\boldsymbol{\beta}^{j,c})^T \boldsymbol{K} (\boldsymbol{\beta}^{j,c}) - \Lambda\right], \end{aligned} \tag{20}$$

for $\gamma, \lambda_j^c, \mu_j^c > 0$.

To find the stationary points, we set the partial derivatives of the Lagrangian to zero,

$$\frac{\partial}{\partial \boldsymbol{\alpha}} J = 2\left(\boldsymbol{M} - \gamma \boldsymbol{N}\right) \boldsymbol{\alpha} - \sum_{c=1}^{C} \sum_{j=1}^{k_c} \lambda_j^c \left(\boldsymbol{K} \boldsymbol{\beta}^{j,c} + (\boldsymbol{M}_- - \boldsymbol{M}_+^c)\right) = 0, \tag{21}$$

$$\frac{\partial}{\partial \boldsymbol{\beta}^{j,c}} J = -\lambda_j^c \left(\boldsymbol{K}\boldsymbol{\alpha}\right) - 2\mu_j^c \boldsymbol{K}\boldsymbol{\beta}^{j,c} = 0, \quad \text{for} \quad j = 1, 2, \dots k_c, \quad c = 1, 2, \dots C. \tag{22}$$

Substituting $\boldsymbol{\beta}^{j,c} = -\frac{\lambda_j^c}{2\mu_j^c}\boldsymbol{\alpha}$, which results from (22), into (21) yields

$$2\left(\boldsymbol{M} - \gamma \boldsymbol{N}\right)\boldsymbol{\alpha} = -\sum_{c=1}^{C}\sum_{j=1}^{k_c} \lambda_j^c \left(\boldsymbol{K}\frac{\lambda_j^c}{2\mu_j^c}\boldsymbol{\alpha} + \left(\boldsymbol{M}_- - \boldsymbol{M}_+^c\right)\right), \tag{23}$$

which can be further simplified as

$$\left(\boldsymbol{M} - \gamma \boldsymbol{N}\right)\boldsymbol{\alpha} = -\boldsymbol{K}\boldsymbol{\alpha}\sum_{c=1}^{C}\sum_{j=1}^{k_c}\frac{(\lambda_j^c)^2}{4\mu_j^c} - \sum_{c=1}^{C}\left\{\left(\boldsymbol{M}_- - \boldsymbol{M}_+^c\right)\sum_{j=1}^{k_c}\frac{\lambda_j^c}{2}\right\}. \tag{24}$$

Let $\omega = -\sum_{c=1}^{C}\sum_{j=1}^{k_c}\frac{(\lambda_j^c)^2}{4\mu_j^c}$, and $\nu^c = -\sum_{j=1}^{k_c}\frac{\lambda_j^c}{2}$. Then, we have

$$\left(\boldsymbol{M} - \gamma \boldsymbol{N}\right)\boldsymbol{\alpha} = \boldsymbol{K}\boldsymbol{\alpha}\omega + \sum_{c=1}^{C}\left\{(\boldsymbol{M}_- - \boldsymbol{M}_+^c)\nu^c\right\}. \tag{25}$$

Therefore, $\boldsymbol{\alpha}$ is the solution of the linear system

$$\left(\left(\boldsymbol{M} - \gamma \boldsymbol{N}\right) - \omega \boldsymbol{K}\right)\boldsymbol{\alpha} = \sum_{c=1}^{C}\left\{(\boldsymbol{M}_- - \boldsymbol{M}_+^c)\nu^c\right\}. \tag{26}$$

Solving the problem yields the optimal projection coefficients $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*]$. Subsequently, we find the projection of a new test point $\boldsymbol{x}_{\text{test}} \subset \mathcal{X}$ onto $\boldsymbol{w}$ by

$$< \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}_{\text{test}}) >= \sum_{\ell=1}^{l} \alpha_\ell^* K(\boldsymbol{x}_\ell, \boldsymbol{x}_{\text{test}}). \tag{27}$$

The solution of the linear system of equations, namely (26), provides us with the coefficients $\boldsymbol{\alpha}^*$ which will be used for finding a tighter boundary for the majority class. We note that despite not being present in (26), $\boldsymbol{\beta}^{j,c}$'s affect the values of $\boldsymbol{\alpha}$ through the values of the Lagrangian coefficients, λ_j^c and μ_j^c. As such, $\boldsymbol{\beta}^{j,c}$'s are used implicitly to identify the locations of absent points, although not explicitly needed for prediction; this is how the use of absent points helps find a tighter boundary for the majority class.

3. Algorithm

As mentioned in Section 2, in optimization problem (15)-(17) we find the projection coefficients $\boldsymbol{\alpha}^*$ based upon two considerations specifically developed to address the imbalanced structure. The outcome is a decision boundary separating the two classes. Yet, those absent data points are still implicitly considered and have not been used to update the estimate of scatter matrices. This section explains how to generate the synthetic data points, based on the newly decided class boundary, and use them to update the scatter matrices.

From a different angle, optimization problem (15)-(17) can be seen as a way to expand the region associated with the minority data points, as opposed to the region one would have had without imposing constraints (16) and (17). This expansion allows us to identify the minority region with better precision and to estimate $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ more accurately. Once the minority class region is revised, we can use the knowledge to synthesize more data points for the minority class.

We start by using an iterative procedure that alternately updates the class boundary and revises the $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ estimation. In other words, optimization problem (15)-(17) splits the input region $\mathcal{X}$ into two disjoint regions $\mathcal{X}^-$ and $\mathcal{X}^+$ which are estimated regions belonging to the majority and minority points, respectively. Then, we draw additional minority points, i.e. data synthesizing for the minority class, from the updated minority region to improve the estimates of the scatter matrices.

Specifically, we draw independent samples from the estimated density of the current minority points, conditional on the boundary imposed by the optimal projection coefficients $\boldsymbol{\alpha}_*$. Let $\widehat{F}^u_{\boldsymbol{\alpha}_*}$ be the estimated distribution of the minority points as a mixture of u Gaussian distribution estimated using $\mathcal{X}^+ = \{\boldsymbol{x}_1^+, \boldsymbol{x}_2^+, \ldots, \boldsymbol{x}_l^+\} \subset \mathcal{X}$ and truncated according to $\boldsymbol{\alpha}_*$, namely

$$\widehat{F}^u_{\boldsymbol{\alpha}_*} = \frac{1}{u} \sum_{b=1}^{u} a_b \Psi_b, \tag{28}$$

where Ψ_b is a Gaussian distribution with mean $\boldsymbol{\mu}_b$ and variance $\boldsymbol{\Sigma}_b^2$, truncated over the region $\mathcal{X}^+$, $0 \leq a_b \leq 1$ for $b = 1, 2, \ldots, u$, and $\sum_{b=1}^{u} a_b = 1$. Let $\widetilde{\mathcal{Z}}$ denote a set of q independent samples drawn from $\widehat{F}^u_{\boldsymbol{\alpha}_*}$, specifically $\widetilde{\boldsymbol{x}}_\ell^+ \sim \widehat{F}^u_{\boldsymbol{\alpha}_*}$, for $\ell = 1, 2, \ldots, q$. Denote the augmented minority set by $\widetilde{\mathcal{X}}^+ = \mathcal{X}^+ \bigcup \widetilde{\mathcal{Z}} = \{\boldsymbol{x}_1^+, \boldsymbol{x}_2^+, \ldots, \boldsymbol{x}_l^+, \widetilde{\boldsymbol{x}}_1^+, \widetilde{\boldsymbol{x}}_2^+, \ldots, \widetilde{\boldsymbol{x}}_q^+\}$. Note the difference between $\widetilde{\boldsymbol{x}}_\ell^+$ used here and $\boldsymbol{z}_\ell$ used in the previous section: $\boldsymbol{z}_\ell$ denotes the absent data points close to the class boundary, playing a role similar to the support vector points, while $\widetilde{\boldsymbol{x}}_\ell^+$ denotes any data point actually generated for the minority class. The $\widetilde{\boldsymbol{x}}_\ell^+$ points cannot be guaranteed to be close to the class boundary; rather they may be over the interior of the minority region or cross the boundary and over the region of the majority class (called intrusion). Consequently, there is a difference between k and q: k is the number of data points represented by $\boldsymbol{z}_\ell$, similar to the number of support vector points, while q is the number of actually generated data points scattering around in the input space. Generally, q is larger than k.

Then we use the augmented minority set to reevaluate the between- and within-class scatter matrices, such as:

$$\widetilde{\boldsymbol{S}}_B^\phi = \left(\boldsymbol{m}_-^\phi - \widetilde{\boldsymbol{m}}_+^\phi\right)\left(\boldsymbol{m}_-^\phi - \widetilde{\boldsymbol{m}}_+^\phi\right)^T,$$

$$\widetilde{\boldsymbol{S}}_W^\phi = \sum_{\boldsymbol{x} \in \mathcal{X}^-} (\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{m}_-^\phi)(\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{m}_-^\phi)^T + \sum_{\boldsymbol{x} \in \widetilde{\mathcal{X}}^+} \left(\boldsymbol{\phi}(\boldsymbol{x}) - \widetilde{\boldsymbol{m}}_+^\phi\right)\left(\boldsymbol{\phi}(\boldsymbol{x}) - \widetilde{\boldsymbol{m}}_+^\phi\right)^T, \tag{29}$$

where $\widetilde{\boldsymbol{m}}_+^\phi = \frac{1}{l+q} \sum_{j=1}^{l+q} \boldsymbol{\phi}(\boldsymbol{x}_j^+)$. In other words, we update the estimates of the scatter matrices using the newly generated points. Using (7) and (8) and following the steps for the optimization procedure stated in Section 2, we obtain a new optimization problem similar to (15)-(17) in which the matrices $\boldsymbol{K}$, $\boldsymbol{N}$, and $\boldsymbol{M}$ and vectors $\boldsymbol{M}_-$ and $\boldsymbol{M}_+^c$, for

$c = 1, 2, \ldots, C$, are evaluated using the sets $\mathcal{X}^-$ and $\widetilde{\mathcal{X}}^+$. The new optimization problem yields a new optimal projection coefficient vector $\boldsymbol{\alpha}_*$ which, in turn, we use to re-estimate the scatter matrices by fitting again a mixture of Gaussian distributions and generating $q \leftarrow \lfloor \frac{q}{2} \rfloor$ absent points (i.e. half of the points we generated in the previous iteration). We continue this procedure until $q < 1$, and we use the final $\boldsymbol{\alpha}_*$ as the optimal projection coefficient vector.

The clusters at each stage are decided based on the X-means algorithm (Pelleg and Moore, 2000). X-means is simply a k-means clustering algorithm in which the number of clusters, which is denoted by C in our algorithm, is decided based on a Bayesian Information Criterion (BIC) (Hastie et al., 2009). We choose X-means because the number of clusters is not known in advance; this number is estimated by X-means based on data; other clustering methods can also be used (Fraley and Raftery, 1998).

The number of Gaussian mixtures to estimate the distribution of the minority points is also decided based on BIC. Specifically, the number of Gaussian mixtures at each iteration is

$$\arg\min_{u \in \mathbb{N}} \text{BIC}\left(\widehat{F}^u_{\boldsymbol{\alpha}_*}\right), \tag{30}$$

where $\mathbb{N}$ is the set of positive integers and

$$\text{BIC}\left(\widehat{F}^u_{\boldsymbol{\alpha}_*}\right) = -2\log(L) + u\log(q),$$

where L is the likelihood of the minority data points, assuming they are random samples from $\widehat{F}^u_{\boldsymbol{\alpha}_*}$.

Once we find the number of Gaussian mixtures, we generate q data points such that those points are sampled from the fitted Gaussian mixture, assuming the current boundary defined by the classifier. Among the q synthetic data points at each stage, we first admit $q' \leq q$ of them based on (27); this step is to discard the synthetic data points that are on the wrong side of the decision boundary. We denote the set of the admitted points by $\widetilde{\mathcal{Z}'}$, which is the final set of the newly generated data points at a given stage.

The data points in $\widetilde{\mathcal{Z}'}$ are then assigned to a cluster $c = 1, 2, \ldots, C$ according to their Euclidean distance to the center of the cluster in the original space. Specifically, for $\widetilde{\boldsymbol{x}}^+_\ell \in \widetilde{\mathcal{Z}'}$, its cluster membership is assigned as

$$c = \arg\min\nolimits_{c' \in \{1,\ldots,C\}} \|\widetilde{\boldsymbol{x}}^+_\ell - \bar{\boldsymbol{x}}^+_{c'}\|, \tag{31}$$

where $\bar{\boldsymbol{x}}^+_{c'} = \frac{1}{l_{c'}} \sum_{\ell=1}^{l_{c'}} \boldsymbol{x}^+_{\ell,c'}$, which is the center of cluster c' in the original space. This gives us q_c new data points for each cluster $c = 1, 2, \ldots, C$, such that $\sum_{c=1}^{C} q_c = q'$.

The values of Lagrangian coefficients λ^j_c and μ^j_c are determined to be inversely proportional to the number of current minority data points in their associated clusters, namely $\lambda^j_c = \frac{\lambda}{l_c}$ and $\mu^j_c = \frac{\mu}{l_c}$. This means that if there are very few data points in a cluster, violating constraints (16) and (17) is more heavily penalized in comparison to the case when there are more data points in that cluster. The number of perceived absent data points in a cluster k_c is also inversely proportional to the current number of data points in that cluster, because a cluster formed by very few data points is not reliable enough to generate many new data points. Note that k_c is the *a priori* number of perceived absent data points in a cluster, while q_c is the actually generated data points belonging to cluster c.

Assuming we know the values of the tuning parameters γ and λ, we can summarize the steps of the Absent Data Generator classifier (ADG) in Algorithm 1. In practice, the aforementioned tuning parameters are determined using cross validation (Hastie et al., 2009). Based on our experiments, ADG is not very sensitive to the number of absent points k, so that it can be simply set to a number between 10 to 15. The number of actual minority data points generated, q, on the other hand, is decided so that the final data set of interest is relatively balanced. Note that the number of newly generated points q is decreasing at each stage.

Algorithm 1 Absent Data Generator for Imbalanced Classification

```
Given $\mathcal{X}^-$ and $\mathcal{X}^+$, evaluate $\boldsymbol{K}$, $\boldsymbol{M}$, $\boldsymbol{N}$, $\boldsymbol{K}_i$, and $\boldsymbol{M}_i$, for $i \in \{-,+\}$ and let $\widetilde{\mathcal{X}}^+ = \mathcal{X}^+$.
repeat
  1. Find $C$ clusters for the augmented minority set $\widetilde{\mathcal{X}}^+$, where $C$ is decided by minimizing the associated BIC.
  2. Choose $\lambda_c^j = \frac{\lambda}{l_c}$, $\mu_c^j = \frac{\mu}{l_c}$, for $j = 1, 2, \dots k_c$, and $k_c$ is chosen proportionally to $\frac{1}{l_c}$, for $c = 1, 2, \dots C$, such that $\sum_{c=1}^{C} k_c = k$.
  3. Let $\omega = -\sum_{c=1}^{C} \sum_{j=1}^{k_c} \frac{(\lambda_j^c)^2}{4\mu_j^c}$, $\nu^c = -\sum_{j=1}^{k_c} \frac{\lambda_j^c}{2}$ and $(\boldsymbol{M}_+^c)_j = \frac{1}{l_c} \sum_{\ell=1}^{l_c} K(\boldsymbol{x}_j, \boldsymbol{x}_{\ell,c}^+)$
  4. Let $\boldsymbol{\alpha}_*$ be the solution of $((\boldsymbol{M} - \gamma \boldsymbol{N}) - \omega \boldsymbol{K})\, \boldsymbol{\alpha} = \sum_{c=1}^{C} \left\{(\boldsymbol{M}_- - \boldsymbol{M}_+^c)\nu^c\right\}$.
  5. Fit a mixture of $u$ normal distributions to $\mathcal{X}^+$ where $u$ provides the smallest BIC in (30).
  6. Generate $q$ data points from the resulting Gaussian mixtures above, say $\widetilde{\mathcal{Z}} = \{\widetilde{\boldsymbol{x}}_1^+, \widetilde{\boldsymbol{x}}_2^+, \dots, \widetilde{\boldsymbol{x}}_q^+\}$.
  7. Utilize $\boldsymbol{\alpha}_*$ according to (27) to test if each $\widetilde{\boldsymbol{x}}_\ell^+$, $\ell = 1, 2, \dots, q$ belongs to class $+1$ or not. Let $\widetilde{\mathcal{Z}'} = \boldsymbol{\{} x_{l+1}^+, \boldsymbol{x}_{l+2}^+, \dots, \boldsymbol{x}_{l+q'}^+ \} \subset \widetilde{\mathcal{Z}}$ be the set of data points admitted into the minority set.
  8. Identify the clusters to which the new data points belong according to (31). Let $q_c$ be the number of elements in $\widetilde{\mathcal{Z}'}$ belonging to cluster $c$, for $c = 1, 2, \dots C$.
  9. $\widetilde{\mathcal{X}}^+ \leftarrow \widetilde{\mathcal{X}}^+ \cup \widetilde{\mathcal{Z}'}$.
  10. $\widetilde{\mathcal{X}} \leftarrow \mathcal{X}^- \cup \widetilde{\mathcal{X}}^+$.
  11. $(\boldsymbol{M}_-)_j \leftarrow \frac{1}{l_-} \sum_{\ell=1}^{l_-} K(\boldsymbol{x}_j, \boldsymbol{x}_\ell^-)$ for $\boldsymbol{x}_j \in \widetilde{\mathcal{X}}$.
  12. $(\boldsymbol{M}_+)_j \leftarrow \frac{1}{l+q'} \sum_{\ell=1}^{l+q'} K(\boldsymbol{x}_j, \boldsymbol{x}_\ell^+)$ for $\boldsymbol{x}_j \in \widetilde{\mathcal{X}}$.
  13. $(\boldsymbol{M}_+^c)_j = \frac{1}{l_c+q_c} \sum_{\ell=1}^{l_c+q_c} K(\boldsymbol{x}_j, \boldsymbol{x}_{\ell,c}^+)$ for $\boldsymbol{x}_j \in \widetilde{\mathcal{X}}$.
  14. $(\boldsymbol{K})_{r,s} \leftarrow K(\boldsymbol{x}_r, \boldsymbol{x}_s)$, for $r, s \in \{1, 2, \dots, n+q'\}$,
      $(\boldsymbol{K}_-)_{r,s} = K(\boldsymbol{x}_r, \boldsymbol{x}_s^-)$ for $r \in \{1, 2, \dots, n+q\}$, $s \in \{1, 2, \dots, l_-\}$,
      $(\boldsymbol{K}_+)_{r,s} = K(\boldsymbol{x}_r, \boldsymbol{x}_s^+)$ for $r \in \{1, 2, \dots, n+q\}$, $s \in \{1, 2, \dots, l+q\}$.
  15. $\boldsymbol{M} \leftarrow (\boldsymbol{M}_- - \boldsymbol{M}_+)(\boldsymbol{M}_- - \boldsymbol{M}_+)$.
  16. $\boldsymbol{N} \leftarrow \sum_{i \in \{-,+\}} \boldsymbol{K}_i (\boldsymbol{I} - \boldsymbol{1}_{l_i}) \boldsymbol{K}_i^T$.
  17. $q \leftarrow \lfloor \frac{q}{2} \rfloor$,
      $l \leftarrow |\widetilde{\mathcal{X}}^+|$,
      $n \leftarrow |\widetilde{\mathcal{X}}|$.
until $q < 1$.
```

Having the optimal projection coefficients $\boldsymbol{\alpha}_*$ which corresponds to the optimal projection vector $\boldsymbol{w}$ in the feature space, we can obtain the prediction for the class labels by classifying the projected values of the data points onto $\boldsymbol{w}$. Let $\boldsymbol{\kappa}_{\boldsymbol{x}}$ be an $n \times 1$ vector of the kernel evaluation between $\boldsymbol{x} \in \mathcal{X}$ and all the training samples and the synthetic minority points generated by Algorithm 1, that is,

$$(\boldsymbol{\kappa}_{\boldsymbol{x}})_\ell = K(\boldsymbol{x}_\ell, \boldsymbol{x}), \quad \forall \boldsymbol{x}_\ell \in \widetilde{\mathcal{X}}. \tag{32}$$

Then assume $C_{\mathcal{T}}$ is a one-dimensional binary classifier, e.g. the Support Vector Machine, trained on the set $\mathcal{T} = \{(h(\boldsymbol{x}_\ell; \boldsymbol{\alpha}_*), y_\ell) : \boldsymbol{x}_\ell \in \widetilde{\mathcal{X}}, \ell = 1, 2, \ldots, n\}$, where y_ℓ is the class label for $\boldsymbol{x}_\ell$, and $h(\boldsymbol{x}_\ell; \boldsymbol{\alpha}_*) = \boldsymbol{\alpha}_*^T \boldsymbol{\kappa}_{\boldsymbol{x}_\ell}$. More precisely, the classifier $C_{\mathcal{T}}$, after training on $\mathcal{T}$, yields a real number as the threshold v_* such that if $h(\boldsymbol{x}; \boldsymbol{\alpha}_*) > v_*$, then the corresponding $h(\boldsymbol{x}; \boldsymbol{\alpha}_*)$ is labeled as $+1$; otherwise -1. Then, the label prediction for a test point $\boldsymbol{x}_t$ using the ADG will be

$$\text{ADG}(\boldsymbol{x}_t) = \begin{cases} +1 & \text{if } h(\boldsymbol{x}_t; \boldsymbol{\alpha}_*) > v_*, \\ -1 & \text{if } h(\boldsymbol{x}_t; \boldsymbol{\alpha}_*) \leq v_*. \end{cases} \tag{33}$$

We note that the ADG's data generation mechanism is based on an iterative method that explores the minority region by data generating constraints that are embedded in the optimization problem. Unlike SMOTE, in ADG, the synthetic data are not necessarily in the convex hull of existing data which could be another advantage for ADG, especially in higher dimensions. Also, ADG acknowledges the significance of the data points close to the boundary and generates synthetic data by utilizing both majority and minority data points.

ADG's computational complexity is of polynomial order. Note that the major operation in the algorithm is solving the system of linear equations (26), i.e. step 4 in the algorithm, since all the other steps involve relatively low computational costs. Particularly, clustering, if solved exactly, has cost $\mathcal{O}(n^{dC+1} \log n)$, where d denotes the dimension here (Inaba et al., 1994); however, we appeal to heuristics to accelerate the process even close to a linear order of complexity in n under mild conditions (Kanungo et al., 2002). Other approaches to implement the X-means algorithm faster are discussed by Pelleg and Moore (2000). Fitting a mixture of u normal distributions requires only $\mathcal{O}(nu^2)$ flops (Verbeek et al., 2003). Using the kernel trick does not impact the complexity of the algorithm, and moreover, the number of iterations is $\mathcal{O}(\log n)$. Hence, from a computational complexity perspective, the algorithm is dominated by (26) which can be solved using an LU decomposition in $\frac{2}{3}n^3 + \mathcal{O}(n^2)$ operations (Trefethen and Bau III, 1997). As such, the complexity of the algorithm is $\mathcal{O}(n^3 \log n)$.

4. Experiments

In this section, we apply the proposed ADG algorithm to a number of data sets, both real and artificial, and compare it to five alternative methods. Three of the methods in comparison are the Cost-Sensitive Support Vector Machine (CS-SVM), Synthetic Minority Over-Sampling Technique (SMOTE) (Chawla et al., 2002) and Borderline-SMOTE (BSMOTE) (Han et al., 2005). CS-SVM is an SVM algorithm (Hastie et al., 2009) modified for the imbalanced classification by imposing a higher cost on minority miss-classification (Elkan, 2001). In CS-SVM, we choose the value of the so-called "box constraint" in SVM to be

$\frac{l+l_-}{2l}$ for positive samples and $\frac{l+l-}{2l_-}$ for negative samples, so that the cost ratio for the two-class misclassification is $\frac{l}{l_-}$. BSMOTE is similar to SMOTE but generates the new data points close to the boundary between the minority and majority classes. In this regard, BSMOTE uses a data synthesizing mechanism closest to that used in the proposed ADG. For both SMOTE and BSMOTE, once the new data points are generated, we can use a KFD algorithm to perform the task of classification on the balanced data set. Thereby, it would be straightforward to compare their performance against the ADG, as ADG also has the KFD as its classifier. The main parameter in SMOTE and B-SMOTE is the amount of oversampling, which is set to the same level as that in ADG, which, in turn, is determined by the value of q as discussed in Section 3.

The aforementioned competing algorithms are selected to compare different data generating mechanisms (SMOTE and BSMOTE) with that of the ADG, and to observe how they perform compared to another school of thought in imbalanced classification, cost-sensitive classification (CSSVM). We therefore present comparison among these algorithms in more detail. As a general principle, we select the parameters in the competing methods based on the recommendations made by the authors of the associated papers, unless otherwise indicated.

To further investigate ADG's viability as a means for imbalanced classification, at the end of this section we also compare the results of ADG with a combination of ensemble learners and undersampling (Wallace et al., 2011), and generating data using a fitted probabilistic distribution for the minority data points (Hempstalk et al., 2008; Liu et al., 2007). The former, referred to as "Under+ENS" hereafter, undersamples the majority data points several times to obtain balanced data sets and then uses a set of ensemble classifiers on the balanced data sets. The latter, referred to as "Prob-Fit" hereafter, fits a probability distribution to the existing minority data points and then generates synthetic data points from that distribution to create balanced data sets which, in turn, are used for classification. The probability distribution used in Prob-Fit, for all the data sets used in this paper, is a mixture of Gaussian distributions.

Concerning the kernel function used in both ADG and SVM (recall SVM is used in CS-SVM, SMOTE and BSMOTE), we use a Radial Basis Function kernel $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-d\|\boldsymbol{x} - \boldsymbol{y}\|^2)$, in which the parameter d is estimated through cross validation. To implement KFD we use the MATLAB package `Statistical Pattern Recognition Tool (STPRtool)` (Franc, 2011). We code ADG, SMOTE, BSMOTE, Under+ENS, and Prob-Fit in MATLAB, and also use the SVM implementation in MATLAB.

The performance measures we are interested in are the false alarm rate and detection power. Specifically, for the test set $\{(\boldsymbol{x}_\ell, y_\ell) | \ell = 1, 2, \ldots, N\}$, we can estimate the false alarm rate and detection power as follows

$$\widehat{\mathrm{FA}} = \frac{1}{N_-} \sum_{\ell=1}^{N_-} \mathcal{L}_{(0,1)}(y_\ell, \hat{y}_\ell), \quad \text{for } \ell \,\text{such that}\, y_\ell = -1, \tag{34}$$

and

$$\widehat{\mathrm{DP}} = 1 - \frac{1}{N_+} \sum_{\ell=1}^{N_+} \mathcal{L}_{(0,1)}(y_\ell, \hat{y}_\ell), \quad \text{for } \ell \,\text{such that}\, y_\ell = +1, \tag{35}$$

where N_- and N_+ are the number of majority and minority points in the test set, respectively. The variable $\hat{y}_i$ is the predicted class label (i.e. -1 or 1) for the associated prediction method, and $\mathcal{L}_{(0,1)}(.,.)$ is the 0-1 loss function

$$\mathcal{L}_{(0,1)}(y_1, y_2) = \begin{cases} 0 & \text{if } y_1 = y_2, \\ 1 & \text{if } y_1 \neq y_2. \end{cases} \tag{36}$$

Concerning the numerical experiments, we need to utilize simulated/real data sets which are deemed imbalanced. However, the number of available imbalanced data sets is limited, and we are also interested in testing algorithms on data sets with varying degrees of imbalance ratio, which can be characterized by the proportion of the majority data points to the minority data points in each data set. To this end, having the original training sets, $\mathcal{X}^+$ and $\mathcal{X}^-$, we can build training sets that are comprised of a subset of $\mathcal{X}^+$ and $\mathcal{X}^-$ and have a different proportion of majority to minority compared to the original training sets. That is, we have $\mathcal{X}_u^+ \subset \mathcal{X}^+$ and $\mathcal{X}_u^- \subset \mathcal{X}^-$ where $\frac{\mathcal{X}_u^+}{\mathcal{X}_u^-} > \frac{\mathcal{X}^+}{\mathcal{X}^-}$. Then we can utilize $\mathcal{X}_u^+$ and $\mathcal{X}_u^-$ as the new training set and the remaining data for testing. We will explain this approach in Section 4.2.

4.1 Using a Simulated Data set

Before presenting the classification results using the real data sets, we want to observe the difference of the mechanism of data generation between ADG and SMOTE. For this purpose, we create one simulated data set, in which we generate 900 data points as the majority data set from a mixture of five Gaussian distributions on $\mathbb{R}^2$ and 450 data points as the minority data set from a mixture of another five Gaussian distributions on $\mathbb{R}^2$.

Figure 1 shows a sample of synthetic data generation for a subset of the mixture of Gaussian distributions with an imbalance ratio greater than 6. Comparing region A in plots (b) and (c) in Figure 1 suggests that, for this particular data set, the ADG mechanism is more "space-filling" than that of SMOTE. Comparing region B in plots (b) and (d) shows that the intrusion into the majority space, while attempting to be space-filling, is less of a problem for ADG than that for BSMOTE, which also aims at generating data close to the boundary. Performing this space-filling property within the minority region, is of paramount importance for imbalanced classification in higher dimensions as well. It is not easy to demonstrate this property for the other data sets, as their dimensions are larger than two. The subsequent numerical results, however, support ADG's potency in imbalanced classification, and we think its strength can be partly attributed to ADG's ability to maintain the property better than SMOTE and BSMOTE.

4.2 Real Data sets

We use a total of eleven real data sets for training and testing. Four of them are from the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/`), which are the Wisconsin Diagnostic Breast Cancer data set, the Ionosphere data set, the Yeast data set and Speech Recognition data set. The other seven are used in (Wallace and Dahabreh, 2012) (`http://www.cebm.brown.edu/static/imbalanced-datasets.zip`). Table 1 summarizes the basic properties associated with these data sets, including the Gaussian mixture data simulated in Section 4.1.

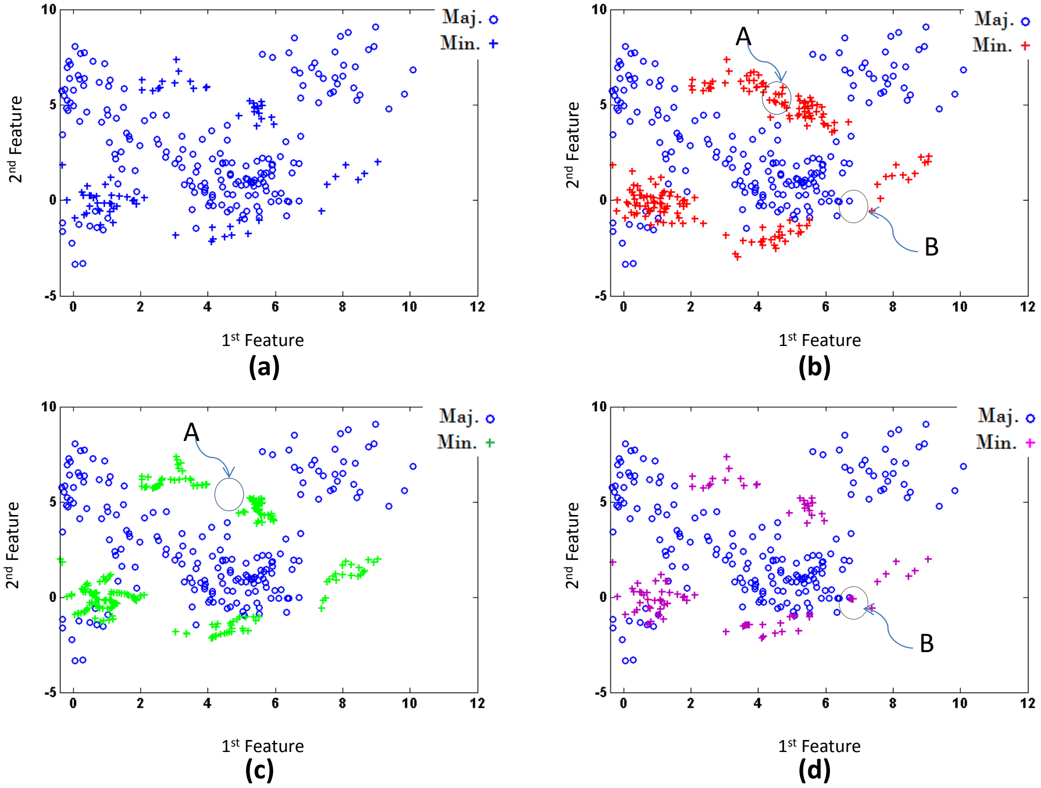

Figure 1: Comparing the mechanism of data generation in ADG with SMOTE for an artificial data set: (a) Original imbalanced data; (b) Balanced data after one iteration of ADG; (c) Balanced data after using SMOTE; (d) Balanced data after using BSMOTE. Comparing region A in plots (b) and (c) and region B in plots (c) and (d) shows ADG is more space-filling and intrudes less into the majority space.

Among the aforementioned data sets, not all of them are genuinely imbalanced. In those circumstances, we form the training data sets using a large portion of the majority data and a very small portion of the minority data. Besides, we are interested in observing how different methods perform as a data set becomes more imbalanced. For this purpose, we adjust the degrees of imbalance in a training set, by tuning the ratio of the number of majority points over the number of minority points in the data set. Specifically, for a given imbalance ratio, we first randomly undersample both the majority and the minority data points so that the training data set is constructed with the specified degree of imbalance. This means we obtain new training sets $\mathcal{X}_u^+$ and $\mathcal{X}_u^-$ as explained in the beginning of this section, run each algorithm on the training set, and use the remaining data for testing. We repeat this procedure ten times and report the average values as the estimated false alarm rate and detection power. Note that these new $\mathcal{X}_u^+$ and $\mathcal{X}_u^-$ will have the role of $\mathcal{X}^+$ and $\mathcal{X}^-$ in Algorithm 1 and no further modification is applied to the algorithm.

Data set	Dimension	Total Data Amount	# of Majority	# of Minority
Simulated Gaussian mixtures	2	1350	900	450
Breast Cancer Detection	9	699	458	241
Speech Recognition	10	990	900	90
Yeast	10	1484	1449	35
Ionosphere	34	351	225	126
Pima	8	768	500	268
Car	21	1728	1659	69
Ecoli	9	336	301	35
Glass	9	214	197	17
Haberman	3	306	225	81
Vehicle	18	846	634	212
CMC	24	1473	1140	333

Table 1: Basic properties of data sets

4.3 Results

We represent the performance of each algorithm on each data set using the Area Under Curve (AUC) of the Receiver Operating Characteristic (ROC) plot (Bradley, 1997). In the ROC analysis we plot each (FA, DP) point for a test case in an ROC space in which the FA is on the x-axis and the DP is on the y-axis (Provost et al., 1997). We use the `perfcurve` command in MATLAB to generate the ROC curves, once we have computed a sufficient number of (FA, DP) points. Then, we compute AUC as the area under a respective ROC curve. Note that a larger AUC generally denotes better performance.

We apply the six competing methods (ADG included) to the twelve data sets (including the simulated Gaussian mixture data) under different imbalance ratios. We report the average AUC and its standard deviation (both from ten repetitions), instead of the ROC plots themselves. Considering the number of classification methods in comparison, data sets involved, and imbalance ratios used, it is impractical to hope that plotting all ROC curves can produce a clear overall picture. Instead, we present the AUC information in a concise form: Table 2 lists the average values and Table 3 lists the corresponding standard deviations.

As evident in Table 2, ADG provides the largest AUC for most cases, especially under the most imbalanced circumstances of each test instance. Rather than expecting the ROC to suggest the optimal classifier, one may identify the regions or scenarios where a classifier can be recommended (Provost et al., 1997). We find that ADG provides a good balance between the conflicting objectives of reducing the false alarm, while increasing the detection power.

As expected, Prob-Fit performs very well on the simulated data, because the data are simulated using Gaussian mixture models. On the real data sets, the performance of Prob-Fit depends on the actual number of minority data points, that is, it performs better when the minority data are enough to reliably fit a distribution, and it performs poorly when the data set suffers from absolute scarcity. Therefore, simply fitting a distribution to generate data is of little use (Liu et al., 2007). The mechanism behind the performance of Under+ENS seems to be more involved, and it appears to be competitive for a few cases only. The comparisons demonstrate the importance of the structure of specific data sets, and that no one classifier is dominant for all types of data under all imbalance ratios. The relation between a data structure and the mechanism embedded in the classifiers to handle the imbalanced data is of interest to be understood, but currently there are not enough insights garnered and we leave that issue to future efforts.

The fact that there are no dominating classifiers leads us to ask whether ADG's performance is statistically significant compared to the other methods. Considering that we are in presence of several classifiers and several data sets, we need to use a test which ranks classifiers based on their performance, followed by a post hoc analysis. One classical method which we utilize is the Friedman test (Demšar, 2006), a non-parametric method which sorts the algorithms conducted on several data sets. Let m^a be the number of algorithms, i.e. classifiers, and m^d be the number of data sets. Let Re be an $m^d \times m^a$ matrix of the results listed in Table 2, in which each row represents a data set and each column is a classifier. Considering the average results for each imbalance ratio as produced by one "data set", we have $m^d = 48$ and $m^a = 6$. First, define the matrix Ra whose entries in each row represent the classifier's rank for that specific data set. Under the null hypothesis that all classifiers are equivalent, i.e. their performance on each data set is identical, the Friedman statistic

$$\mathcal{F} = \frac{12m^d}{m^a(m^a+1)} \left(\sum_{\ell=1}^{m^a} \overline{Ra}_\ell^2 - \frac{m^a(m^a+1)^2}{4} \right), \tag{37}$$

has a Chi-squared distribution with $m^a - 1$ degrees of freedom, where $\overline{Ra_\ell}$ is the average value of column $\ell = 1, 2, \ldots, m^a$. Table 4 lists the means for the estimated ranks associated with each method. Figure 2, which presents the post hoc analysis on the ranking data using multiple comparisons, shows the ADG's ranking is significantly higher than other competing algorithms under the 0.05 level of significance.

Before concluding this section, we want to briefly discuss the drawbacks of the cost-sensitive approach (Maloof, 2003) and one-class classification (also known as novelty detection) (Park et al., 2010). One major obstacle faced with cost-sensitive methods is how to choose a suitable cost ratio that leads to robust outcomes. Figure 3 shows the detection power and false alarm as a function of cost ratio for the Haberman data where an imbalance ratio greater than 3 is used in training. Specifically, the cost ratio denotes the value associated with the box constraint in the SVM for minority data points divided into that value for the majority data points. As Figure 3 shows, the detection power remains almost constant after the cost ratio passes a threshold around 7, yet the false alarm rate continues to increase. Similar evidence has been documented in the literature regarding the lack of robustness in choosing a good cost ratio in the cost-sensitive methods (Byon et al., 2010). This lack of robust performance is one reason why synthetic oversampling is generally more powerful than cost-sensitive methods.

Some researchers favor one-class classification (OCC) approaches to solve imbalanced data problems. In other words, it is better to ignore the data points due to their sparseness in the minority data set, and instead create a closed decision boundary to characterize the majority data only. In a detection mission, one would classify a new data point as belonging either to the majority or the minority class. This OCC approach can be useful for some extreme cases in which the number of data points in the minority is so few that there are no practical ways to elicit any relevant information. In many practical cases, however, despite

Data	Imb. Ratio	ADG	SMOTE	BSMOTE	CSSVM	Under+ENS	Prob-Fit
Gaussian Mixture	7	**0.886**	0.879	0.879	0.886	0.601	0.879
	4	0.888	0.886	0.881	0.888	0.675	**0.903**
	3	0.885	0.887	0.878	0.890	0.659	**0.912**
	2	0.892	0.900	0.886	0.893	0.666	**0.906**
Breast Cancer	6	**0.900**	0.896	0.897	0.895	0.814	0.882
	4	**0.899**	0.893	0.894	0.894	0.856	0.889
	3	**0.905**	0.901	0.902	0.899	0.879	0.900
	2	0.899	0.897	0.897	0.894	0.903	**0.916**
Speech Recognition	29	**0.894**	0.877	0.868	0.871	0.663	0.860
	15	**0.911**	0.900	0.908	0.906	0.774	0.902
	10	0.891	0.898	0.903	0.891	0.867	**0.915**
	7	0.925	0.919	0.921	0.897	**0.932**	0.909
Yeast	121	**0.811**	0.709	0.731	0.760	0.614	0.778
	65	**0.820**	0.723	0.755	0.775	0.683	0.789
	40	**0.849**	0.766	0.812	0.801	0.765	0.810
	27	0.858	0.780	0.807	0.809	0.825	**0.859**
Ionosphere	6	**0.896**	0.890	0.884	0.891	0.796	0.854
	4	**0.891**	0.885	0.878	**0.891**	0.841	**0.891**
	3	0.895	0.888	0.881	0.894	0.869	**0.905**
	2	0.899	0.892	0.885	0.893	0.906	**0.918**
Pima	6	0.681	0.622	0.679	0.668	0.680	**0.718**
	4	0.710	0.660	0.697	0.692	0.702	**0.729**
	3	**0.721**	0.687	0.699	0.692	0.709	0.720
	2	0.734	**0.753**	0.724	0.700	0.709	0.736
Car	69	**0.890**	0.872	0.875	0.889	0.851	0.597
	37	0.898	0.888	0.891	0.896	**0.917**	0.756
	23	0.900	0.895	0.897	0.899	**0.970**	0.873
	15	0.904	0.897	0.903	0.903	**0.991**	0.900
Ecoli	25	**0.729**	0.641	0.696	0.619	0.724	0.681
	14	0.732	0.616	0.705	0.601	**0.773**	0.697
	8	0.731	0.702	0.701	0.613	**0.849**	0.715
	6	0.752	**0.797**	0.681	0.699	0.775	0.722
Glass	33	0.713	0.653	0.669	**0.718**	0.667	0.710
	19	**0.754**	0.716	0.663	0.709	0.649	0.693
	11	**0.779**	0.737	0.768	0.728	0.701	0.729
	8	0.826	**0.896**	0.852	0.796	0.808	0.774
Haberman	8	0.602	0.568	0.549	0.518	0.595	**0.608**
	4	**0.640**	0.543	0.584	0.569	0.586	0.601
	3	**0.653**	0.582	0.573	0.598	0.605	0.625
	2	**0.681**	0.596	0.584	0.596	0.618	0.627
Vehicle	9	**0.714**	0.693	0.708	0.712	0.700	0.701
	5	**0.729**	0.707	0.728	0.729	0.701	0.709
	3	0.783	0.778	0.763	**0.831**	0.712	0.729
	2	0.782	0.796	0.790	**0.843**	0.770	0.735
CMC	10	0.589	0.532	0.538	0.586	**0.607**	0.549
	5	**0.679**	0.593	0.593	0.664	0.639	0.555
	3	0.682	0.646	0.667	**0.712**	0.652	0.605
	2	0.692	0.683	0.678	**0.727**	0.670	0.641

Table 2: Average Area Under Curve (AUC). The largest values in each row are boldfaced. "Imb. Ratio" means imbalance ratio, the ratio of the number of majority points over the number of minority points in a data set.

Data	Imb. Ratio	ADG	SMOTE	BSMOTE	CSSVM	Under+ENS	Prob-Fit
Gaussian Mixture	7	0.053	0.034	0.037	0.032	0.061	0.012
	4	0.065	0.037	0.037	0.037	0.061	0.008
	3	0.048	0.047	0.050	0.048	0.050	0.008
	2	0.020	0.016	0.017	0.024	0.023	0.009
Breast Cancer	6	0.008	0.013	0.012	0.009	0.009	0.015
	4	0.015	0.011	0.012	0.013	0.015	0.008
	3	0.011	0.010	0.010	0.012	0.012	0.010
	2	0.013	0.011	0.020	0.013	0.012	0.009
Speech Recognition	29	0.021	0.021	0.018	0.018	0.021	0.018
	15	0.044	0.033	0.029	0.033	0.041	0.021
	10	0.027	0.042	0.020	0.034	0.031	0.010
	7	0.035	0.020	0.023	0.032	0.033	0.012
Yeast	121	0.028	0.039	0.045	0.029	0.029	0.046
	65	0.042	0.055	0.044	0.032	0.039	0.046
	40	0.070	0.075	0.067	0.063	0.073	0.069
	27	0.165	0.163	0.154	0.135	0.165	0.152
Ionosphere	6	0.038	0.032	0.030	0.036	0.037	0.028
	4	0.032	0.029	0.027	0.034	0.039	0.030
	3	0.028	0.031	0.020	0.029	0.028	0.013
	2	0.024	0.023	0.022	0.019	0.023	0.022
Pima	6	0.026	0.022	0.021	0.031	0.030	0.017
	4	0.031	0.037	0.023	0.034	0.033	0.016
	3	0.019	0.020	0.021	0.021	0.023	0.018
	2	0.026	0.023	0.027	0.028	0.027	0.022
Car	69	0.033	0.040	0.050	0.033	0.033	0.054
	37	0.025	0.074	0.068	0.031	0.028	0.120
	23	0.015	0.024	0.028	0.015	0.016	0.037
	15	0.006	0.040	0.043	0.005	0.006	0.082
Ecoli	25	0.060	0.082	0.073	0.070	0.070	0.089
	14	0.075	0.091	0.087	0.073	0.090	0.085
	8	0.045	0.051	0.049	0.039	0.047	0.058
	6	0.154	0.144	0.138	0.140	0.144	0.130
Glass	33	0.083	0.094	0.096	0.088	0.098	0.092
	19	0.114	0.118	0.114	0.108	0.134	0.109
	11	0.150	0.174	0.113	0.149	0.155	0.126
	8	0.146	0.138	0.156	0.132	0.161	0.133
Haberman	8	0.041	0.040	0.040	0.042	0.043	0.037
	4	0.053	0.057	0.043	0.049	0.060	0.029
	3	0.045	0.055	0.064	0.046	0.053	0.054
	2	0.049	0.053	0.053	0.043	0.049	0.050
Vehicle	9	0.016	0.019	0.017	0.017	0.019	0.017
	5	0.025	0.027	0.026	0.029	0.028	0.027
	3	0.027	0.024	0.024	0.029	0.029	0.019
	2	0.033	0.051	0.042	0.027	0.031	0.054
CMC	10	0.026	0.048	0.057	0.023	0.025	0.080
	5	0.016	0.038	0.049	0.016	0.019	0.060
	3	0.020	0.021	0.022	0.020	0.024	0.022
	2	0.073	0.089	0.076	0.079	0.079	0.088

Table 3: Standard deviation for Area Under Curve (AUC) reported in Table 2.

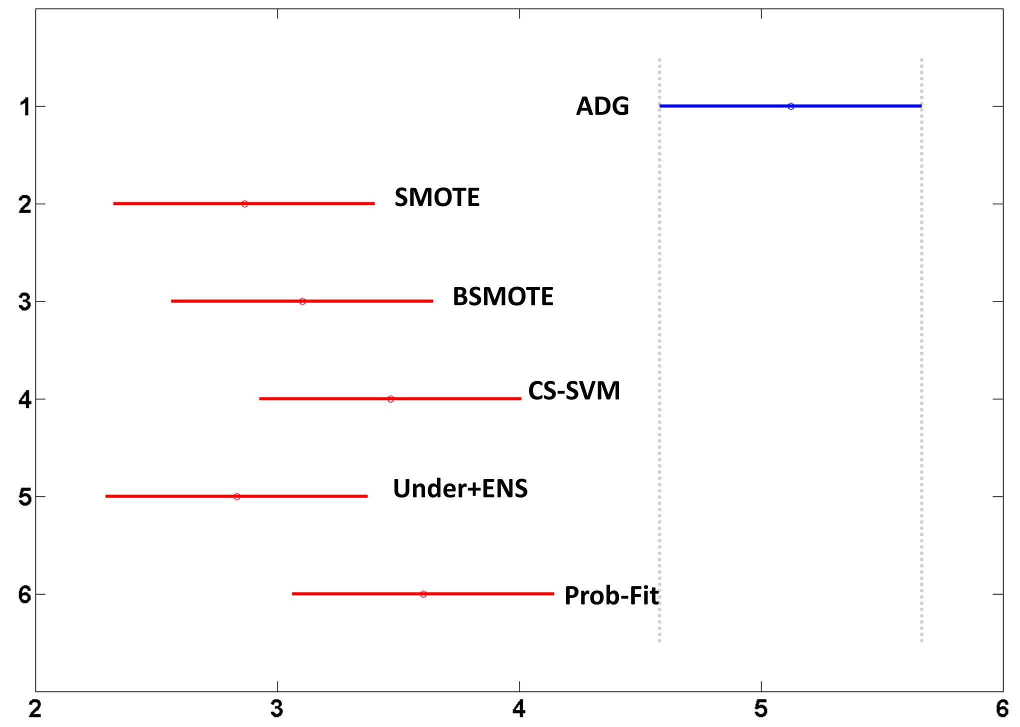

Figure 2: Post hoc analysis on the ranking data obtained by the Friedman test. ADG's mean column rank is significantly higher than other classifiers.

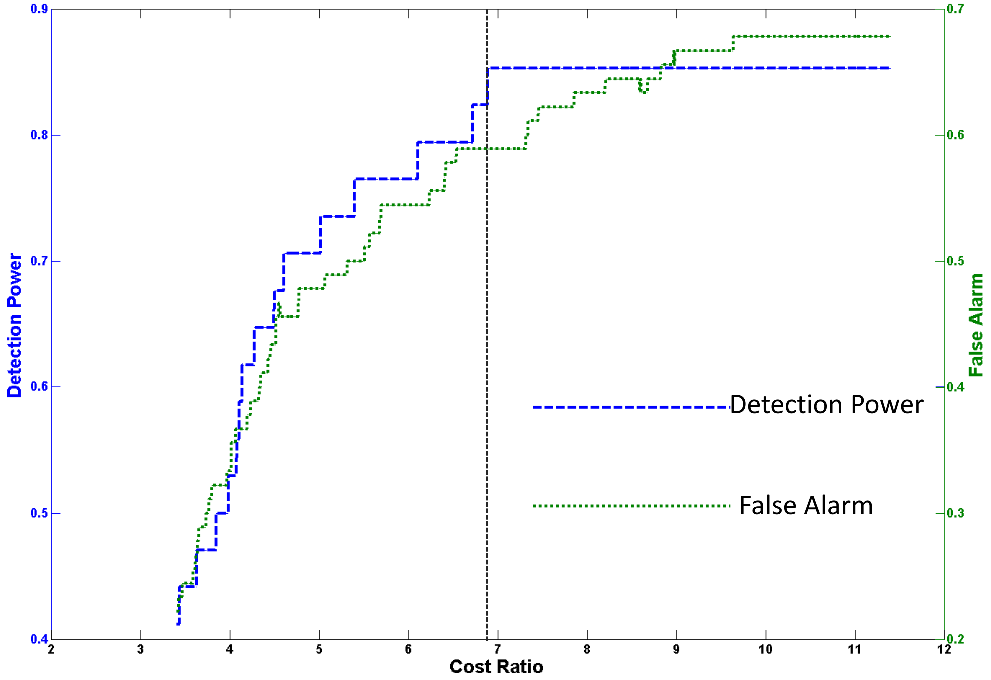

Figure 3: Detection power (left axis) and false alarm (right axis) as a function of the cost ratio in CS-SVM for the Haberman data set.

Classifier	ADG	SMOTE	BSMOTE	CSSVM	Under+ENS	Prob-Fit
Mean of Ranking	5.125	2.865	3.104	3.469	2.833	3.604

Table 4: Mean of rankings based on Friedman test

the sparseness of the data, minority data sets still can provide useful information if utilized appropriately. To demonstrate the usefulness of utilizing the minority data, we compare ADG with the OCC method developed in Park et al. (2010) using four sample data sets; this OCC method was proven to provide asymptotically the tightest bound for majority data points. For these four sample data sets, we select the training and test data such that the training data sets have the smallest value of imbalance ratio reported in Table 2. As Figure 4 shows, the OCC could be effective, for instance, duplicating ADG's performance in the case of Pima data. One drawback is that OCC methods often suffer from a high false alarm rate, while attaining a high detection power (e.g. in the case of the Ionosphere data). When an OCC tries to build the tightest possible closed boundary around the majority data, the result can be an over-tightened boundary, instead of a boundary loose enough to identify all majority data points. On the other hand, in the two-class cases, the existence of minority data points can actually help relax the position of the decision boundary, at least locally where these minority data points are present. For more detailed comparisons of another OCC method with two-class classifiers, the reader may consult (Hempstalk et al., 2008); the results presented there also confirm the argument that if minority data are utilized, one generally observes an improvement in the minority detection.

5. Extension and Error Bounds

In this section, we consider two additional aspects regarding the proposed algorithm. First, we seek to identify bounds on the generalization error for the ADG. Second, we extend the proposed method to deal with the multi-class classification in which a subset of classes has very few observations available in the training stage.

5.1 Bounds on Generalization Error

Generalization error refers to the expected error on test instances coming from the same distribution of the training sample (Rasmussen and Williams, 2006). Specifically, if $\boldsymbol{x} \sim \mathcal{G}$, where $\mathcal{G}$ is the distribution of the input $\boldsymbol{x}$, the generalization error of some decision function h with respect to loss function $\mathcal{L}$ is defined as

$$\mathbb{E}_{\boldsymbol{x}}\{\mathcal{L}(h)\}, \tag{38}$$

where $\mathbb{E}$ is the expectation operator.

Let $\boldsymbol{\alpha}_F$ denote the optimal value of $\boldsymbol{\alpha}$ obtained by solving optimization problem (3), namely the KFD. Similar to the procedure explained in Section 3 for obtaining the prediction label for ADG, let $C_{\mathcal{U}}$ be the same one-dimensional binary classifier used for ADG, trained on the set $\mathcal{U} = \{(h(\boldsymbol{x}_\ell; \boldsymbol{\alpha}_F), y_\ell) : \boldsymbol{x}_\ell \in \mathcal{X}^- \cup \mathcal{X}^+, \ell = 1, 2, \ldots, n\}$, where $\boldsymbol{\kappa}_{\boldsymbol{x}}$ is defined similarly to (32) for $\boldsymbol{x}_\ell \in \mathcal{X}^- \cup \mathcal{X}^+$, and $h(\boldsymbol{x}_\ell; \boldsymbol{\alpha}_F) = \boldsymbol{\alpha}_F^T \boldsymbol{\kappa}_{\boldsymbol{x}_\ell}$. If the threshold value for the

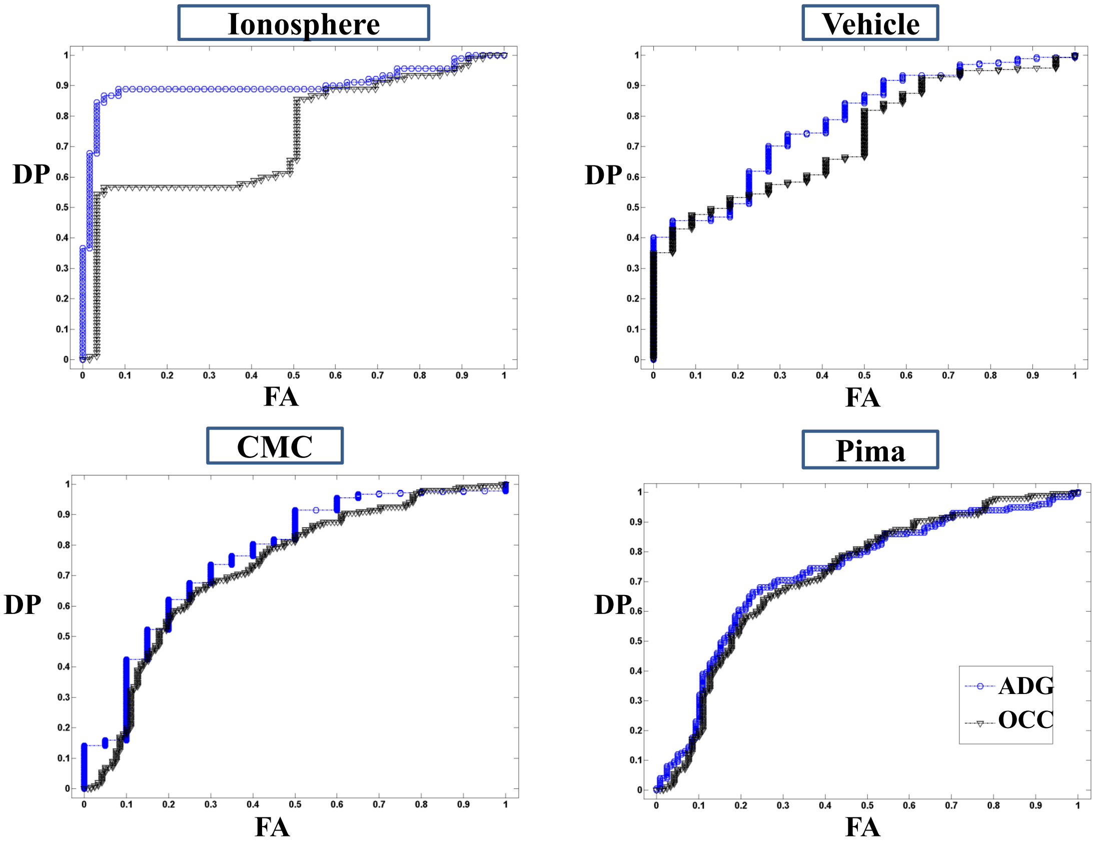

Figure 4: Comparing ROCs for ADG and OCC for four sample data sets.

$C_{\mathcal{U}}$ is v_F, we have the following prediction for a test point $\boldsymbol{x}_t$ using the KFD

$$\text{KFD}(\boldsymbol{x}_t) = \begin{cases} 1 & \text{if } h(\boldsymbol{x}_t; \boldsymbol{\alpha}_F) > v_F, \\ -1 & \text{if } h(\boldsymbol{x}_t; \boldsymbol{\alpha}_F) \leq v_F. \end{cases} \tag{39}$$

Consequently, following the total law of probability, we can deduce that the generalization error of KFD is equal to

$$err_{\text{K}} = \pi_- \mathbb{P}\left[h(\boldsymbol{x}_t; \boldsymbol{\alpha}_F) > v_F | y_t = -1\right] + \pi_+ \mathbb{P}\left[h(\boldsymbol{x}_t; \boldsymbol{\alpha}_F) \leq v_F | y_t = 1\right], \tag{40}$$

where π_i is the prior probability that a point belongs to the class $i \in \{-, +\}$.

Durrant and Kabán (2012) established an upper bound on this generalization error, under the assumption that the data points of each class follow a Gaussian distribution once mapped to the feature space. Specifically, having a training data set of size $n = l_+ + l_-$ and assuming data in the feature space are normally distributed with mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}$ for $i \in \{-, +\}$, then for any $\rho \in (0, 1)$ the generalization error of KFD is bounded above with probability of at least $1 - \rho$ by $ub(l, \rho)$ where

$$ub(l, \rho) = \sum_{i \in \{-,+\}} \pi_i \Phi \left(-2 \left[g(\bar{\tau}(\epsilon)) \times \Pi - \sqrt{\frac{n}{l_i}} \left(1 + \sqrt{\frac{2}{n} \log \frac{4}{\rho}} \right) \right] \right), \tag{41}$$

where

$$\Pi = \left[\sqrt{\frac{\|\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-\|^2}{\lambda_{\max}(\boldsymbol{\Sigma})} + \frac{n}{l_- l_+} \frac{\text{tr}(\boldsymbol{\Sigma})}{\lambda_{\max}(\boldsymbol{\Sigma})}} - \sqrt{\frac{2n}{l_- l_+} \log \frac{4}{\rho}} \right]_+, \tag{42}$$

$g(r) = \frac{\sqrt{r}}{1+r}$ for $r \in \mathbb{R}$, $\lambda_{\max}(\boldsymbol{\Sigma})$ is the largest eigenvalue of the covariance matrix, $[.]_+ = \max(0, .)$, Φ is the CDF of the standard normal distribution, and

$$\bar{\tau}(\epsilon) = \frac{\lambda_{\max}(\boldsymbol{\Sigma})}{\eta} \left(1 + \sqrt{\frac{n-2}{n}} + \frac{\epsilon}{\sqrt{n}} \right)^2 + \tau(\boldsymbol{\Sigma}_n), \tag{43}$$

where $\epsilon = \sqrt{2 \log \frac{4}{\rho}}$, $\tau(\boldsymbol{\Sigma}_n)$ denotes the condition number of $\boldsymbol{\Sigma}_n$ that is the covariance matrix of the points in a subset of the feature space generated by the n points in $\mathcal{X}^- \cup \mathcal{X}^+$, and η is a regularization constant to ensure non-singularity of the estimate of $\boldsymbol{\Sigma}_n$. As $g(.)$ is a monotonic decreasing function on $r \geq 1$, a smaller value for $\bar{\tau}(\epsilon)$ suggests a smaller value for the upper bound. Note that assuming the regularization constant η does not need to change as more data points are added to the training set, then the only quantities which affect $\bar{\tau}(\epsilon)$ are $\tau(\boldsymbol{\Sigma}_n)$ and n.

Note that as the number of observations increases, (41) yields a tighter bound, assuming that all other quantities remain constant. This is in fact what happens in synthetic data generation, especially for ADG, since it generates extra observations at each iteration of the algorithm. The more subtle issue is how the estimated value of the covariance matrix $\boldsymbol{\Sigma}$, projected in the Hilbert space generated by the observation, changes with the generation of more data points.

Note that $\boldsymbol{\Sigma}_n = \boldsymbol{P} \boldsymbol{\Sigma} \boldsymbol{P}^T$, where $\boldsymbol{P}$ is an orthogonal projection into the Hilbert space spanned by the observations. Assuming that data points mapped to the feature space are linearly independent, we can have $\boldsymbol{P}_n = \left(\boldsymbol{X}_n^{\phi T}\boldsymbol{X}_n^{\phi}\right)^{-\frac{1}{2}}\boldsymbol{X}_n^{\phi T}$, where $\boldsymbol{X}_n^{\phi}$ is a matrix whose columns are $\boldsymbol{\phi}(\boldsymbol{x}_\ell)$ for $\boldsymbol{x}_\ell \in \mathcal{X}^- \cup \mathcal{X}^+$. If we add a new observation $\boldsymbol{x}_{n+1}$ to the training set $\mathcal{X}^- \cup \mathcal{X}^+$, we will get the projection matrix $\boldsymbol{P}_{n+1}^{\boldsymbol{\phi}}$. See the Appendix for an explanation that as the number of data points increases, the condition number of the covariance matrix of the space generated by the data points in the feature space decreases, which in turn implies we achieve a tighter bound for generalization error using ADG.

In fact, as long as a synthetic data generation mechanism is embedded in a KFD framework, as ADG does, we can invoke the above theoretical result on the reduction of the generalization error. Despite the fact that SMOTE and BSMOTE can also be used, note that their data generation mechanisms cannot be integrated with KFD. For this reason, the above error bound result cannot be readily applied to SMOTE and BSMOTE.

5.2 Extension to Multi-class Classification

The methodology presented for the imbalanced two-class classification can be easily extended to cover multi-class classification in which a subset of classes lack sufficient observations for the training stage. Let $\mathcal{X}^i = \{\boldsymbol{x}_1^i, \boldsymbol{x}_2^i, \ldots, \boldsymbol{x}_{l_i}^i\} \subset \mathcal{X}$ denote the training set for class $i \in \mathcal{I} = \{1, 2, \ldots, I_s\}$, where $l_{i_2} \ll l_{i_1}$, for $i_1 \in \mathcal{I}_1$, $i_2 \in \mathcal{I}_2$, where $\mathcal{I}_1 \cup \mathcal{I}_2 = \mathcal{I}$ and $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$. Let $\mathcal{Z}_{i_s} = \{\boldsymbol{x}_{l_{i_s}+1}^{i_s}, \boldsymbol{x}_{l_{i_s}+2}^{i_s}, \ldots, \boldsymbol{x}_{l_{i_s}+k_{i_s}}^{i_s}\} \subset \mathcal{X}$ be the absent data from the minority class i_s, and denote each $\boldsymbol{x}_{l_{i_s}+k_{i_s}}$ by $\boldsymbol{z}_j^{i_s}$. For simplicity, consider a case in which the data in each group consist of a single cluster, i.e. $C = 1$; however, the following algorithm can be readily extended to consider more clusters. Assume that the data are centered around each covariate so they have mean 0. Sequentially solve the following optimization problem to obtain $\boldsymbol{w}_i$ for $i \in \mathcal{I}$:

$$\max_{\boldsymbol{w}_i} J(\boldsymbol{w}_i) = \frac{\boldsymbol{w}_i^T \boldsymbol{S}_B^{\boldsymbol{\phi}} \boldsymbol{w}_i}{\boldsymbol{w}_i^T \boldsymbol{S}_W^{\boldsymbol{\phi}} \boldsymbol{w}_i}, \tag{44}$$

subject to

$$\boldsymbol{w}_i \perp \boldsymbol{w}_\ell, \quad \forall \ell < i, \tag{45}$$

$$\left(\boldsymbol{w}_i^T \boldsymbol{\phi}(\boldsymbol{z}_j^{i_s}) - \boldsymbol{w}^T \boldsymbol{m}_{i_s}^{\boldsymbol{\phi}}\right)^2 \leq \delta, \tag{46}$$

$$(\boldsymbol{\phi}(\boldsymbol{z}_j^{i_s}) - \boldsymbol{m}_{i_d}^{\boldsymbol{\phi}})^T(\boldsymbol{\phi}(\boldsymbol{z}_j^{i_s}) - \boldsymbol{m}_{i_r}^{\boldsymbol{\phi}}) \leq \Lambda \quad \text{for} \quad j = 1, 2, \ldots k_{i_s}, \quad i_s \in \mathcal{I}_2, i_r \in \mathcal{I}_1, \tag{47}$$

where $\boldsymbol{S}_B^{\boldsymbol{\phi}}$ and $\boldsymbol{S}_W^{\boldsymbol{\phi}}$ are the between and within class scatter matrices, respectively, in the feature space

$$\boldsymbol{S}_B^{\boldsymbol{\phi}} = \sum_{i \in \mathcal{I}} l_i \boldsymbol{m}_i^{\boldsymbol{\phi}} (\boldsymbol{m}_i^{\boldsymbol{\phi}})^T,$$

$$\boldsymbol{S}_W^{\boldsymbol{\phi}} = \sum_{i \in \mathcal{I}} \sum_{\boldsymbol{x} \in \mathcal{X}^i} (\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{m}_i^{\boldsymbol{\phi}})(\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{m}_i^{\boldsymbol{\phi}})^T, \tag{48}$$

and $\boldsymbol{m}_i^{\phi} = \frac{1}{l_i}\sum_{j=1}^{l_i} \boldsymbol{\phi}(\boldsymbol{x}_j^i)$, for $i \in \mathcal{I}$. For each minority class in $\mathcal{I}_2$, generate k_{i_s} artificial points from class $i_c \in \mathcal{I}_2$. Similar to the two-class classification problem, use the Representer's Theorem to replace each $\boldsymbol{w}_i$ and $\boldsymbol{\phi}(\boldsymbol{z}_j^{i_s}) - \boldsymbol{m}_{i_r}^{\boldsymbol{\phi}}$ as linear combinations of the training data in the feature space as in (7) and (8). This leads to systems of linear equations as in (26) which can be embedded into an algorithm similar to Algorithm 1.

6. Summary

This paper presents an algorithm for solving the two-class classification with imbalanced training data. The difficulty associated with such data structures is that the inadequate number of data points belonging to one class (i.e. minority) leads to the problem that most two-class classification algorithms tend to favor the majority class in labeling test points. To solve the problem, we devise an algorithm that relies on minority data synthesis. At each iteration we solve an optimization which considers more numbers of minority points without explicitly specifying them. Those points affect our decision by forcing the algorithm to set the decision boundary as though the points genuinely existed. We draw samples from the new region to enable a more accurate estimation for the scatter matrices. Using several simulated and real data sets, we compare the performance of the resulting ADG algorithm with the competing methods, CS-SVM, SMOTE, BSMOTE, Under+ENS and Prob-Fit. The results suggest that using ADG is preferable when there is a pronounced data imbalance.

This paper is a first step for developing a data mechanism embedded in a classification algorithm which we proved useful based on empirical evidence. Since the introduction of SMOTE (Chawla et al., 2002), there has been significant attention to synthetic data generation. We suggest however, that more research is needed to understand the relationship between data generation and classification algorithms.

There are a few critical issues which deserve further attention in this regard. First, the impact of the data structure on the data generation mechanism needs to be studied more thoroughly. The current procedure of data generation may not be suitable for all data structures. Certain alterations on the algorithm, based on the knowledge of how the physical system of interest works, can help improve the performance of ADG. Second, ADG can benefit from an investigation into certain assumptions made in the algorithm. One place is on the assumption that the absent data reside in existing clusters. While reasonable, it might be restrictive for some data sets. Another aspect is that in the current iteration of algorithm, we eliminate all artificial data points that fall on the majority side; this appears beneficial in the examples we studied. Whether or not it can be beneficial for all types of data remains unclear. These issues are certainly important and how to address them is an ongoing pursuit.

Acknowledgments

Arash Pourhabib, Yu Ding and Bani K. Mallick were partially supported by grants from NSF (DMS-0914951, CMMI-0926803, and CMMI-1000088) and King Abdullah University of Science and Technology (KUS-CI-016-04). The authors are also grateful of the valuable suggestions made by the editor and referees that greatly improved the paper.

Appendix A.

We want to show as the number of training data points increases, the condition number of the projected covariance matrix into the Hilbert space generated by the data points decreases. Let $\boldsymbol{x}_\ell \in \mathcal{X}$ for $\ell = 1, 2, \ldots, n$ denote the data points in the original space and let

$\boldsymbol{\zeta}_\ell$ for $\ell = 1, 2, \ldots, n$ denote the data points mapped to a separable Hilbert space $\mathcal{H}$ using a feature map $\boldsymbol{\phi}$, that is $\boldsymbol{\zeta}_\ell = \boldsymbol{\phi}(\boldsymbol{x}_\ell)$ for $\ell = 1, 2, \ldots, n$. Suppose $\mathcal{H}_n$ is an n dimensional subspace of $\mathcal{H}$ spanned by $\boldsymbol{\zeta}_\ell$ for $\ell = 1, 2, \ldots, n$. If $\boldsymbol{\zeta}_\ell$ follow a normal distribution in $\mathcal{H}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, we can have $\boldsymbol{\Sigma}_n$ as the projected covariance matrix into the finite dimensional space $\mathcal{H}_n$. More precisely, $\boldsymbol{\Sigma}_n = \boldsymbol{P}_n \boldsymbol{\Sigma} \boldsymbol{P}_n^T$, namely $\boldsymbol{P}_n$ is an orthogonal projection into $\mathcal{H}_n$, where $\boldsymbol{P}_n = \left(\boldsymbol{X}_n^{\phi T} \boldsymbol{X}_n^{\phi}\right)^{-\frac{1}{2}} \boldsymbol{X}_n^{\phi T}$ and $\boldsymbol{X}_n^{\phi} = [\boldsymbol{\zeta}_\ell : \ell = 1, 2, \ldots, n]$. We want to show that the condition number of $\boldsymbol{\Sigma}_n$ is larger than or equal to that of $\boldsymbol{\Sigma}_{n+1}$.

Without loss of generality, after a rotation and scaling of the data, assume $\left(\boldsymbol{X}_p^{\phi T} \boldsymbol{X}_p^{\phi}\right) = \boldsymbol{I}$, for $p \in \mathbb{N}$, where $\boldsymbol{I}$ is the identity matrix of appropriate size. Therefore,

$$\boldsymbol{P}_{n+1} = \left(\boldsymbol{X}_{n+1}^{\phi}\right)^T = \left[(\boldsymbol{X}_n^{\phi})^T | \boldsymbol{\zeta}_{n+1}^T\right], \tag{49}$$

and

$$\lambda\text{max}(\boldsymbol{\Sigma}_{n+1}) = \lambda\text{max}\left(\boldsymbol{P}_{n+1} \boldsymbol{\Sigma} \boldsymbol{P}_{n+1}^T\right) = \lambda\text{max}\left(\begin{bmatrix} (\boldsymbol{X}_n^{\phi})^T \\ \boldsymbol{\zeta}_{n+1}^T \end{bmatrix} \boldsymbol{\Sigma} \left[\boldsymbol{X}_n^{\phi} | \boldsymbol{\zeta}_{n+1}\right]\right) \tag{50}$$

$$= \lambda\text{max}\left(\begin{bmatrix} \boldsymbol{\Sigma}_n & (\boldsymbol{X}_n^{\phi})^T \boldsymbol{\Sigma} \boldsymbol{\zeta}_{n+1} \\ \boldsymbol{\zeta}_{n+1}^T \boldsymbol{\Sigma} \boldsymbol{X}_n^{\phi} & \boldsymbol{\zeta}_{n+1}^T \boldsymbol{\Sigma} \boldsymbol{\zeta}_{n+1} \end{bmatrix}\right). \tag{51}$$

Let $\|\boldsymbol{\zeta}_{n+1}\|^2 := \boldsymbol{\zeta}_{n+1}^T \boldsymbol{\Sigma} \boldsymbol{\zeta}_{n+1}$. Therefore,

$$\lambda\text{max}(\boldsymbol{\Sigma}_{n+1}) \leq \lambda\text{max}(\boldsymbol{\Sigma}_n) + \|\boldsymbol{\zeta}_{n+1}\|^2, \tag{52}$$

and

$$\lambda_{\text{min}}(\boldsymbol{\Sigma}_{n+1}) \geq \lambda_{\text{min}}(\boldsymbol{\Sigma}_n) + \|\boldsymbol{\zeta}_{n+1}\|^2. \tag{53}$$

Let $\tau(.)$ denote the condition number of a matrix, so

$$\tau(\boldsymbol{\Sigma}_{n+1}) = \frac{\lambda\text{max}(\boldsymbol{\Sigma}_{n+1})}{\lambda_{\text{min}}(\boldsymbol{\Sigma}_{n+1})} \leq \frac{\lambda\text{max}(\boldsymbol{\Sigma}_n) + \|\boldsymbol{\zeta}_{n+1}\|^2}{\lambda_{\text{min}}(\boldsymbol{\Sigma}_n) + \|\boldsymbol{\zeta}_{n+1}\|^2} < \tau(\boldsymbol{\Sigma}_n). \tag{54}$$

References

Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. Applying support vector machines to imbalanced datasets. In *Proceedings of the fifteenth European Conference on Machine Learning (ECML)*, pages 39–50. Springer, 2004.

Kurt Anstreicher and Henry Wolkowicz. On Lagrangian relaxation of quadratic matrix constraints. *SIAM Journal on Matrix Analysis and Applications*, 22(1):41–55, 1998.

Gustavo E.A.P.A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. 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, 2004.

Peter J. Bickel and Elizaveta Levina. Some theory for Fishers linear discriminant function, naive Bayes, and some alternatives when there are many more variables than observations. *Bernoulli*, 10:989–1010, 2004.

Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

Eunshin Byon, Abhishek K. Shrivastava, and Yu Ding. A classification procedure for highly imbalanced class sizes. *IIE Transactions*, 42(4):288–303, 2010.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

J. J. Chen, C. A. Tsai, J. F. Young, and R. L. Kodell. Classification ensembles for unbalanced class sizes in predictive toxicology. *SAR and QSAR in Environmental Research*, 16(6):517–529, 2005.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

Robert J. Durrant and Ata Kabán. Error bounds for kernel Fisher linear discriminant in Gaussian Hilbert space. In *Proceedings of the fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*, volume 22, pages 337–345, 2012.

Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 973–978, 2001.

Chris Fraley and Adrian E Raftery. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

Vojtech Franc. *The Statistical Pattern Recognition Toolbox, Version 2.11*. 2011. URL `http://cmp.felk.cvut.cz/cmp/software/stprtool/index.html`.

Alexander Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 148–155, 1998.

Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing*, volume 3644 of *Lecture Notes in Computer Science*, pages 878–887. Springer Berlin Heidelberg, 2005.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2009.

Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

Kathryn Hempstalk, Eibe Frank, and Ian H. Witten. One-class classification by combining density and class probability estimation. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5211 of *Lecture Notes in Computer Science*, pages 505–519. Springer Berlin Heidelberg, 2008.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220, 2008.

Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. In *Proceedings of the Tenth Annual Symposium on Computational Geometry*, pages 332–339. ACM, 1994.

Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI)*, pages 111–117, 2000.

Tapas Kanungo, David M Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (7):881–892, 2002.

Alexander Liu, Joydeep Ghosh, and Cheryl E. Martin. Generative oversampling for mining imbalanced datasets. In *International Conference on Data Mining*, pages 66–72, 2007.

Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(2):539–550, 2009.

Marcus A. Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *Proceedings of ICML-2003 Workshop on Learning from Imbalanced Data Sets II*, 2003.

Sebastian Mika, Gunnar Rätsch, Jason Weston, Bernhard Schölkopf, and Klaus-Robert Müllers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, pages 41 –48, Aug 1999.

Chiwoo Park, Jianhua Z. Huang, and Yu Ding. A computable plug-in estimator of minimum volume sets for novelty detection. *Operations Research*, 58(5):1469–1480, Sep 2010.

Dan Pelleg and Andrew Moore. X-means: Extending K-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 727–734. Morgan Kaufmann, 2000.

Foster Provost, Tom Fawcett, and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 445–453. Morgan Kaufmann, 1997.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Kai Ming Ting. An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):659–665, 2002.

Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, 1997.

Jakob J. Verbeek, Nikos Vlassis, and Ben Kröse. Efficient greedy learning of Gaussian mixture models. *Neural Computation*, 15(2):469–485, 2003.

Byron C. Wallace and Issa J. Dahabreh. Class probability estimates are unreliable for imbalanced data (and how to fix them). In *IEEE Twelfth International Conference on Data Mining (ICDM)*, pages 695–704, 2012.

Byron C. Wallace, Kevin Small, Carla E. Brodley, and Thomas A. Trikalinos. Class imbalance, redux. In *IEEE Eleventh International Conference on Data Mining (ICDM)*, pages 754–763, 2011.

Gary M. Weiss. Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter-Special Issue on Learning from Imbalanced Datasets*, 6(1):7–19, 2004.