CGMOS: Certainty Guided Minority OverSampling

Paper ID: 588

Abstract

Handling imbalanced datasets is a challenging problem that if not treated correctly results in reduced classification performance. Imbalanced datasets are commonly handled using minority oversampling, whereas the SMOTE algorithm is a successful oversampling algorithm with numerous extensions. SMOTE extensions do not have a theoretical guarantee during training to work better than SMOTE and in many instances their performance is data dependent. In this paper we propose a novel extension to the SMOTE algorithm with a theoretical guarantee for improved classification performance. The proposed approach considers the classification performance of both the majority and minority classes. In the proposed approach CGMOS (Certainty Guided Minority OverSampling) new data points are added by considering certainty changes in the dataset. The paper provides a proof that the proposed algorithm is guaranteed to work better than SMOTE for training data. Further experimental results on 30 real-world datasets show that CGMOS works better than existing algorithms when using 6 different classifiers.

1 Introduction

In many real world problems, the distribution of data between classes is imbalanced. Learning from imbalanced datasets is an important research problem with many applications.

The fundamental issue in imbalanced learning is the ability of imbalanced data to significantly compromise the performance of standard learning algorithms [He and Garcia, 2009]. Generally, there are three primary reasons that can cause this problem [Weiss, 2004].

The first reason is that the lack of data in the minority class makes it difficult to detect regularities within the minority class. Thus, the learned decision boundaries are less likely to approximate the true decision boundaries.

Second, many classification algorithms utilize a general bias for better generalization and to avoid overfitting during learning. However, such bias can adversely affect the ability to learn the minority class. Inductive bias also plays a key role with respect to the minority class. Most classification algorithms prefer more common classes in the presence of uncertainty (i.e., they are biased in favor of the class priors).

Last but not least, noise exerts a greater impact on the minority class, because in this case it is more difficult for a classifier to distinguish noise from minority data. This is especially so in extreme cases where the number of noisy samples is greater than actual minority samples. The problem of overfitting rises again, when modifying the classifier to learn the minority data correctly.

To address these problems, numerous research efforts have been devoted to imbalanced learning in recent years. The majority of techniques that solve the imbalanced learning problem fall into two categories: cost-sensitive methods and sampling-based methods. In the next section, we review related work on sampling-based methods.

1.1 Related work

A number of solutions to the class-imbalance problem were previously proposed both at the data and algorithmic levels [Chawla *et al.*, 2004]. There are mainly three groups of methods that can solve imbalanced learning problem [He and Garcia, 2009] including sampling methods, cost sensitive methods, and kernel methods. Sampling-based methods are very effective and easy to use when solving imbalanced learning problems. In addition, sampling-based methods can be used together with methods in the other two groups to further improve performance. In such approaches a sampling technique is used to modify an imbalanced dataset to produce a balanced distribution. It has been shown that for most imbalanced datasets, sampling techniques do improve classification accuracy.

The basic sampling methods include undersampling and oversampling. Undersampling reduces majority class samples while oversampling increases minority class samples. While several works achieving data balance through undersampling have been proposed in the past [Liu *et al.*, 2009][Zhang and Mani, 2003], more research efforts have been devoted to oversampling due to the fact that oversampling does not discard information.

The simplest form of oversampling is duplication of minority class samples. This approach decreases the overall level of class imbalance, but may lead to overfitting [Drummond and Holte, 2003]. SMOTE [Chawla *et al.*, 2002] is a fundamental approach for oversampling using data synthesis. To balance the dataset, SMOTE randomly selects a seed sample and synthesizes a new sample by applying a linear interpolation between the seed sample and one of its neighbors. Large research efforts have been devoted to feature space data synthesis based on SMOTE. Several methods integrate data synthesis as a part of the learning procedure. For example, by introducing SMOTE in each iteration of boosting, SMOTEBoost [Chawla *et al.*, 2003] increases the number of minority class samples and focus on these cases in each boosting iteration. Using the same idea of boosting, DataBoost-IM [Guo and Viktor, 2004] and RAMOBoost [Chen *et al.*, 2010] discover samples difficult to classify during each iteration of boosting, which are used to guide the oversampling in both the majority and minority classes.

In another group of minority oversampling approaches, the data synthesis procedure is independent of the learning processes. Such methods give preferences to different regions of a dataset by assigning weights to samples in the dataset. These weights can then generate a probability distribution which is used for randomly drawing samples. In such approaches the data synthesis can be completed in one step. Methods in this group include Borderline-SMOTE [Han *et al.*, 2005], Adasyn [He *et al.*, 2008], [Barua *et al.*, 2011] and MWMOTE [Barua *et al.*, 2014]. All of these methods synthesize more samples along decision boundaries. However, these methods do not have objective functions to systematically guide the process of oversampling and so do not have a systematic way to decide on where new data should be synthesized. Thus, such approaches cannot measure the impact of each synthetic sample. As a result, there are several potential problems. One is that the oversampling procedure may sacrifice the performance of the majority class in order to improve the performance of the minority class in the classification. Another is that synthetic minority samples themselves can be misclassified and affect the performance in the minority class.

1.2 Novel Contribution

The proposed approach, CGMOS, is a member of the SMOTE family that can achieve data oversampling in a single step. To address some of the shortcomings in existing approaches, we propose a novel oversampling strategy by systematically considering the performance of both minority and majority classes. Based on a Bayesian classification framework, our proposed approach computes the influence of minority data addition on the certainty of the entire dataset. CGMOS thus can synthesize new samples that will improve the overall certainty of the entire dataset in classification. We prove that during training CGMOS is guaranteed to perform better than SMOTE when using Bayesian classification. To validate the proof, We further show experimentally that CGMOS outperforms known approaches when tested on real-world data set collections using different classifiers.

2 Problem Formulation

In this paper, we address the binary classification problem for imbalanced datasets. Let $D = \{(x_j, y_j)\}_{j=1}^n$ be a training dataset, where $x_j \in \Re^m$ are features and $y_j \in \{l_{\text{mjr}}, l_{\text{mnr}}\}$ are ground truth class labels. We begin by formally defining the certainty of imbalanced binary classification using a Bayesian framework, where a kernel density estimation (KDE) is used to estimate the samples' probability density function (PDF). We then show how CGMOS can synthesize more samples according to the certainty estimation.

2.1 Definition of Certainty

Suppose (x_j, y_j) is any tuple in the training dataset D, where x_j is a feature vector and y_j is the ground truth label of x_j.

A Bayesian classifier maps $x_j \to l, l \in \{l_{\text{mjr}}, l_{\text{mnr}}\}$ using following rule.

$$l = \begin{cases} l_{\text{mnr}} & \text{if } \frac{P(l_{\text{mnr}}|x_j)}{P(l_{\text{mjr}}|x_j)} > 1 \\ l_{\text{mjr}} & \text{otherwise} \end{cases}$$

where the posterior probability $P(l|x_j)$ is computed using Bayes' rule:

$$P(l|x_j) = \frac{P(x_j|l)P(l)}{P(x_j)}; \quad l \in \{l_{\text{mjr}}, l_{\text{mnr}}\}$$

Uncertainty is commonly used in machine learning algorithms. In this work, we use the posterior probability $P(y_j|x_j)$ to define certainty. This is because in classification, the posterior probabilities $P(y_j|x_j)$ reflect the certainty of assigning a sample to a correct label, where higher numbers indicate classification results with a stronger certainty.

Definition 1. (Certainty) Let (x_j, y_j) be any tuple in D, where x_j is a feature vector and y_j is the ground truth label of x_j. The certainties for samples in the majority and minority class are respectively defined as:

$$C(y_j = l_{\text{mjr}}|x_j) = P(y_j = l_{\text{mjr}}|x_j) \tag{1}$$

$$C(y_j = l_{\text{mnr}}|x_j) = P(y_j = l_{\text{mnr}}|x_j) \tag{2}$$

It should be noted that in the case of binary classification the definition of certainty above is related up to some constants to the uncertainty defined in [Sharma and Bilgic, 2013] based on margin confidence.

2.2 PDF Estimation

There are two general ways to estimate a density function: parametric or non-parametric. In this work we use a non-parametric model so as to not depend on a specific distribution model. We use kernel density estimation (KDE)[Elgammal *et al.*, 2002][Zhang *et al.*, 2006] to estimate the likelihood $P(x_j|l), l \in \{l_{\text{mjr}}, l_{\text{mnr}}\}$.

Assuming that the data is independent and identically distributed (i.i.d) and drawn from some distribution with an unknown density $P(x_j|l)$, we have using KDE:

$$P(x_j|l) = \frac{\sum_{k=1}^n K(\frac{x_j - x_k}{h_k})\mathrm{I}(y_k = l)}{\sum_{k=1}^n \mathrm{I}(y_k = l)} \tag{3}$$

where $l \in \{l_{\text{mjr}}, l_{\text{mnr}}\}$, $I(\cdot)$ is an indicator function, and $K(\cdot)$ is a kernel function which has zero mean and integrates to one. Given any sample x_k, the bandwidth h_k of the sample x_k controls the effective range of the kernel and smoothness of the density function. Intuitively one wants to choose h_k as small as the data allows to exhibit as many underlying structures of the data as possible. Small bandwidth, however, will result in a noisy estimate. In this work, for any sample x_k, we calculate a bandwidth h_k as a scaled average distance between x_k and its q nearest neighbors:

$$h_k = \sigma \cdot \frac{\sum_{x \in N(x_k)} \|x - x_k\|}{q} \tag{4}$$

where $N(x)$ is the set of the q nearest neighbors of x_k and $\sigma > 0$ is a scale factor applied to the distance. We will discuss selection of parameters σ and q in Section 4.

2.3 Oversampling Seed Selection

In most classification algorithms, samples close to decision boundaries have less certain classification results. In order to achieve better predictions for such samples, many existing approaches synthesize data directly along the boundaries. However, this is risky and the expected performance improvement is not guaranteed. There are two primary reasons. First, samples from both classes are mixed in regions near the boundaries. Synthetic samples if added to these regions are less predictable and hard to learn. Second, adding synthetic minority samples to these regions may adversely impact the majority class, which may in turn decrease the performance of the majority class in classification. Instead of unguided oversampling near the boundaries, our proposed approach targets adding samples by considering the certainties of both the minority and majority classes before and after adding the samples. The synthetic samples thus are added to locations that can improve the overall certainty of the original data and boost the performance of the classification.

CGMOS uses a similar procedure as SMOTE when synthesizing a new sample. The sample is produced by interpolating between one seed sample and some of its neighbors. However, instead of randomly drawing a seed sample for interpolation, CGMOS assigns each sample $(x_i, y_i) \in D$ a weight $W(x_i)$ which is used to determine the probabilities of x_i being chosen for interpolation. A higher weight results in a higher probability of a point being selected.

To compute $W(x_i)$, we suppose that a new sample will be added to the same location as x_i. The weight $W(x_i)$ is computed as a relative certainty change[1] comparing the certainty before and after the sample is added. With a new sample added at location x_i, we update the certainty for all $(x_j, y_j) \in D$ and denote it as $C_{+i}(y_j|x_j)$.

Definition 2. **(Relative Certainty Change)** The relative certainty change of label y_j assigned to feature x_j due to adding a minority example at location x_i is defined by:

$$R_{+i}(y_j|x_j) = \frac{C_{+i}(y_j|x_j) - C(y_j|x_j)}{C(y_j|x_j)} \tag{5}$$

where $C(y_j|x_j)$ is the certainty before addition.

When computing $W(x_i)$, CGMOS considers the relative certainty changes of examples from both the majority and the minority classes. $W(x_i)$ is computed as the average value of relative certainty changes of all samples in the dataset.

$$W(x_i) = 1 + \frac{1}{n}\sum_{j=1}^{n} R_{+i}(y_j|x_j) \tag{6}$$

Given $W(x_i)$ for all $x_i \in D$, it is easy to see $W(x_i) > 0$. We compute a normalization factor z so that $\frac{1}{z}\sum_{i=1}^{n} W(x_i) = 1$. Therefore, the oversampling procedure can randomly choose sample for interpolation according $W(x_i)/z$. The interpolation phase of CGMOS is the same as SMOTE[Chawla *et al.*, 2002].

A demonstration of CGMOS is shown in Fig. 1. In this figure, samples in both the majority and minority classes are randomly drawn based on Gaussian distribution, where the means of the two datasets are on the same horizontal line, and the mean of the majority is to the right of the minority. The majority class contains 2000 samples and the minority class contains 400 samples. Color in part 1 of the figure indicates the certainty of each example with respect to its class, where red indicates high certainty. We highlight 3 regions (A, B, C) in the minority class. Samples in region A have relative high certainties, sample in region B has low certainties and region C is a boundary region in which samples have the lowest certainties. Part 2 of the figure shows the weight of each example as computed by our approach where red indicates high values. Region B has higher values and is where CGMOS will synthesize most of the samples.

To show the certainty changes induced by adding samples at different locations of the dataset, in part 3 of the figure we add one minority sample and move its location with a fixed step size from left to right on a horizontal line passing through the two classes. We then compute the relative certainty changes for all samples in both classes. As can be observed, by measuring relative certainty changes, CGMOS will assign a higher weight to samples in region B. The figure also shows that by oversampling more in region B, the certainty of the entire dataset gets improved, because the relative certainty changes are positive.

3 Theoretical Guarantee Over SMOTE

Several existing approaches claim handling imbalanced learning better than SMOTE. Such claims are normally validated using empirical tests without a theoretical guarantee and in some instances may not extend to new datasets. In this section we provide a theoretical guarantee showing that CGMOS is expected to work better than SMOTE in training process.

Let $D = \{(x_j, y_j)\}_{j=1}^{n}$ be a training dataset. Let $W(D) = \{W(x_i)\}_{i=1}^{n}$ be the sample weights computed using Eqn. 6.

[1] Measuring absolute certainty increments will not work, because measuring magnitude will give higher preference to parts which already have high certainty.

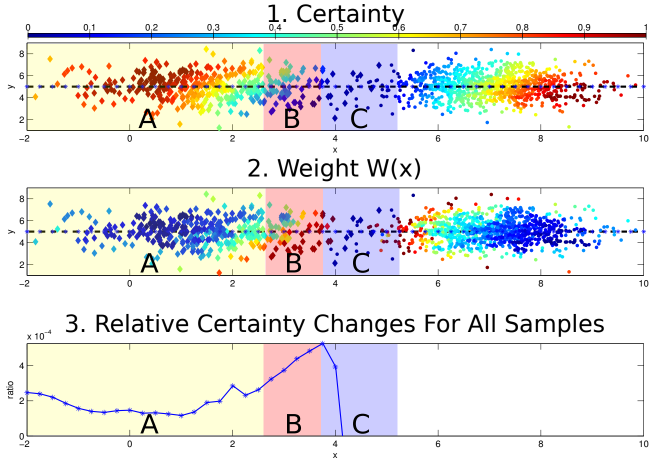

Figure 1: Demonstration of CGMOS. In first two figures, diamonds represent minority samples and circles represent majority samples. The positions of synthesized data points are labeled using a star symbol on a horizontal line passing through the center. The x and y axes represent features. In the bottom figure the x axis indicates a location where a sample was added (in correspondence with the first two figures) whereas the y-axis indicates the relative certainty change.

Lemma 1. *Given a set of weights* $\{W(x_i)\}_{i=1}^n$ *as defined above and a normalization factor* z *given by* $z = \sum_{i=1}^n W(x_i)$*, it must be that* $\sum_{i=1}^n W(x_i)^2 \geq \frac{z^2}{n}$.

Proof Let W be an n-dimensional vector whose elements are $W(x_i)$. Let I be an n-dimensional vector whose elements are all 1. Using the Cauchy-Schwarz inequality we have: $|W \cdot I| \leq \|W\| \cdot \|I\|$. Thus, $|\sum_{i=1}^n W(x_i)| \leq \sqrt{\sum_{i=1}^n W(x_i)^2}\sqrt{n}$ using the fact that $\sum_{i=1}^n W(x_i) = z$, we thus have $\sum_{i=1}^n W(x_i)^2 \geq \frac{z^2}{n}$. ■

Definition 3. (Addition Likelihood Ratio) Let θ denote the non-parametric likelihood estimate $P(x_j|l), l \in \{l_{\text{mjr}}, l_{\text{mnr}}\}$ before a new sample x_i is added, and θ' denote the non-parametric likelihood estimate after the new sample is added. The addition likelihood ratio $r_{+i}(y_j|x_j)$ of example x_j by adding data to x_i location is defined as the ratio between the likelihood estimate after the new addition and the likelihood estimate before the new addition:

$$r_{+i}(y_j|x_j) \equiv P(y_j|x_j;\theta')/P(y_j|x_j;\theta). \tag{7}$$

Lemma 2. *The addition likelihood ratio* $r_{+i}(y_j|x_j)$ *is related to the relative certainty change ratio* $R_{+i}(y_j|x_j)$ *by:*

$$r_{+i}(y_j|x_j) = 1 + R_{+i}(y_j|x_j). \tag{8}$$

Proof According to the definition of the certainty, we have $C_{+i}(y_j|x_j) = P(y_j|x_j;\theta')$ and $C(y_j|x_j;\theta) = P(y_j|x_j;\theta)$. Then $P(y_j|x_j;\theta') = r_{+i}(y_j|x_j)P(y_j|x_j;\theta)$ according to the definition of likelihood ratio. Given Eqn. 5, we have that $R_{+i}(y_j|x_j) = \frac{r_{+i}(y_j|x_j)P(y_j|x_j;\theta) - P(y_j|x_j;\theta)}{P(y_j|x_j;\theta)}$. By simplifying this equation, we thus have

$$r_{+i}(y_j|x_j) = 1 + R_{+i}(y_j|x_j). \quad ■ \tag{9}$$

The addition likelihood ratio defined in Eqn. 7 measures the gain in adding a new point, where higher gains are desired. Note that while the gain is normally close to 1 it may be bigger or smaller than 1.

Definition 4. (Average gain) The average gain when adding sample x_i is defined by:

$$\bar{r}_{+i} = \frac{1}{n}\sum_{j=1}^{n} r_{+i}(y_j|x_j) \tag{10}$$

Lemma 3. *Given the average gain, it must be that:*

$$\bar{r}_{+i} = W(x_i). \tag{11}$$

Proof Using the definition of $W(x_i)$ we have $W(x_i) = \frac{1}{n}\sum_{j=1}^n R_{+i}(y_j|x_j)$. Using Lemma 2 we can replace $r_{+i}(y_j|x_j) - 1$ with $R_{+i}(y_j|x_j)$. Hence:

$$\bar{r}_{+i} = \frac{1}{n}\sum_{j=1}^{n} R_{+i}(y_j|x_j) + 1 \equiv W(x_i) \quad ■ \tag{12}$$

The average gain is an indicator of the benefit of CGMOS. We show that the expected average gain is higher in proposed approach compared with SMOTE.

Theorem 1. *The expected average gain in CGMOS is higher or equal to that of SMOTE.*

Proof For CGMOS the expected average gain is given by:

$$E_p \equiv E[\bar{r}_{+i}] = \sum_{i=1}^{n} \bar{r}_{+i}\frac{W(x_i)}{z} \tag{13}$$

where z is the normalization factor as defined earlier. Using Lemma 3:

$$E_p = \sum_{i=1}^{n} W(x_i)\frac{W(x_i)}{z} = \frac{1}{z}\sum_{i=1}^{n} W^2(x_i). \tag{14}$$

For SMOTE the expected average gain is given by:

$$E_s \equiv E[\bar{r}_{+i}] = \sum_{i=1}^{n} \bar{r}_{+i}\frac{1}{n} \tag{15}$$

Using Lemma 3:

$$E_s = \frac{1}{n}\sum_{i=1}^{n} W(x_i) = \frac{z}{n} \tag{16}$$

Using Lemma 1:

$$E_p = \frac{1}{z}\sum_{i=1}^{n} W^2(x_i) \geq \frac{1}{z}\frac{z^2}{n} = E_s \quad ■ \tag{17}$$

4 Results and Discussion

4.1 Datasets

30 real-world datasets were randomly chosen from the UCI machine learning repository [Lichman, 2013] for empirical testing of CGMOS. Most of the datasets were released within the past 10 years. As some of the datasets contain samples of more than two classes, we convert such datasets to a binary classification problem by keeping the class with the least data and merging all other classes. A summary of the test collections is provided in Table 1.

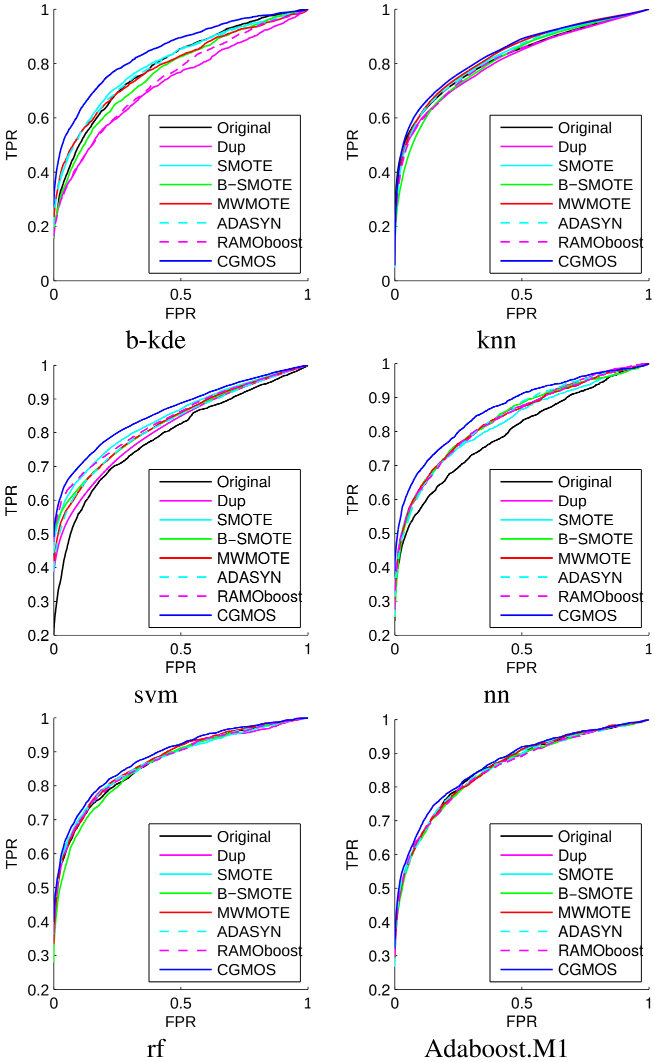

Figure 2: ROC curves of classification results. From left to right, up to down, we show the results of 6 different classifiers: b-kde, knn, svm, nn, rf and Adaboost.M1. Curves in blue are the results of the proposed CGMOS.

4.2 Compared Approaches

According to a survey of imbalanced learning [He and Garcia, 2009], there are mainly three groups of methods addressing imbalanced learning: sampling methods, cost sensitive methods, and kernel methods. The proposed CGMOS belongs to the sampling group. Thus, we compare CGMOS to five other oversampling methods in this group: SMOTE[Chawla *et al.*, 2002], Borderline-SMOTE[Han *et al.*, 2005], ADASYN[He *et al.*, 2008], MWMOTE[Barua *et al.*, 2014] and RAMOBoost[Chen *et al.*, 2010]. Since oversampling by duplication is broadly used in many applications as a baseline, we add it to our evaluation as well. To demonstrate the improvement of these oversampling strategies, we include in the comparison raw data with no oversampling. It should be noted that sampling methods are often combined with cost sensitive methods and kernel methods to further boost learning. [Chawla *et al.*, 2004][Chawla *et al.*, 2003][Guo and Viktor, 2004].

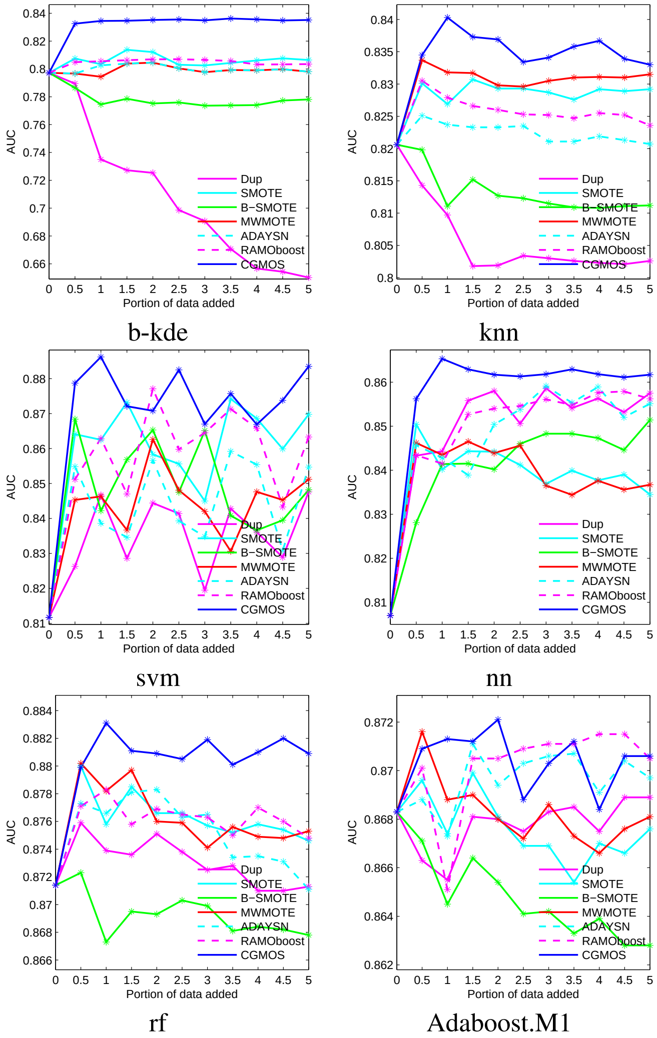

Figure 3: Comparison of results when increasing the number of data synthesized for the minority class. The curves measure the average AUC of the ROC curves. Curves in blue are the results of the proposed CGMOS.

Name	S #	F #	R	Year	Name	S #	F #	R	Year
BankMarket	45211	17	0.13	2012	**Libras**	360	91	0.07	2009
BloodService	748	5	0.31	2008	**MultipleFs**	2000	649	0.11	1998
BreastCancer	400	9	0.53	1988	**Parkinson**	1040	26	0.02	2014
BreastTissue	106	10	0.15	2010	**PlanRelax**	182	13	0.4	2012
CarEvaluation	1730	6	0.04	1997	**QSAR**	1055	41	0.51	2013
Card'graphy	2126	23	0.09	2010	**SPECT**	268	22	0.26	2001
CharacterTraj	2860	3	0.04	2008	**SPECTF**	134	44	0.26	2001
Chess	3198	22	0.91	1989	**SeismicBumps**	2584	19	0.07	2013
ClimateSim	540	18	0.09	2013	**Statlog**	2310	19	0.17	1990
Contraceptive	1474	9	0.29	1997	**PlatesFaults**	1941	27	0.03	2010
Fertility	100	10	0.14	2013	**TAEvaluation**	151	5	0.49	1997
Haberman	306	3	0.36	1999	**UKnowledge**	403	5	0.1	2013
ILPD	580	10	0.4	2012	**Vertebral**	310	6	0.48	2011
ImgSeg	2310	19	0.17	1990	**Customers**	440	8	0.48	2014
Leaf	342	16	0.24	2014	**Yeast**	1484	8	0.04	1996

Table 1: Summary of the datasets used in our experiments, where S#, F#, and R stand for the number of samples, the number of features, and imbalance ratio (defined as #minority/#majority).

4.3 Base classifiers

We match the compared classifiers to classifiers used in other SMOTE extension evaluations. Six well-known classifiers are tested in experiments. The first is the Bayesian classifier based on kernel density estimation described in Section 2 (b-kde). The second is a K nearest neighbors classifier (knn). The third is a support vector machine classifier using RBF kernel (svm). The fourth one is a neural network (nn) with one hidden layer. We use in addition two ensemble methods: a random forest implementing the C4.5 decision tree [Quinlan, 1993] (rf) and Adaboost.M1 [Freund and Schapire, 1996]. All hyper-parameters of the classifiers tested were determined by cross validation to ensure the best performance of each method.

4.4 Evaluation metric

Finding an appropriate evaluation metric for different tasks is challenging, since different evaluation metrics are designed for different purposes. The datasets used in this paper cover from financial application to medical treatment. To achieve an general evaluation and avoid bias, we follow the method in [Chawla *et al.*, 2002][Han *et al.*, 2005][He *et al.*, 2008][Barua *et al.*, 2014][Chen *et al.*, 2010] and use different metrics to evaluate the performance of the proposed CGMOS oversampling algorithm.

Among these evaluation metrics, the most frequently adopted ones are $Precision$ and $Recall$ when the focus of evaluation is focus on one specific class such as problems in text classification, information extraction, natural language processing and bioinformatics. In these areas of application the number of examples belonging to one class is often substantially lower than the overall number of examples, which basically are imbalance learning problems. $Precision$ and $Recall$ are defined as:

$$Precision = \frac{TP}{(TP + FP)}$$

$$Recall = \frac{TP}{(TP + FN)}$$

However, these two metrics share an inverse relationship between each other. A quick inspection on the $Precision$ and $Recall$ formulas readily yields that solely use each of these two metrics only provide a limit view of an algorithm under test. As $Recall$ provides no insight to how many examples are incorrectly labeled as positive and $Precision$ cannot assert how many positive examples are labeled incorrectly. Specifically, the $F\text{-}score$ combines $Precision$ and $Recall$ as measure of the effectiveness of classification in terms of a ration of the weighted importance on either $Recall$ or $Precision$, which is defined as:

$$F\text{-}score = \frac{(1 + \beta^2) \cdot Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall}.$$

We use $\beta = 1$ to treat $Precision$ and $Recall$ equally in all evaluations. As a result, $F\text{-}score$ provides more insight into the functionality of a classifier.

As $F\text{-}score$ measures the harmonic mean of $Precision$ and $Recall$, we also compute $Gscore$ which is the geometric mean of $Precision$ and $Recall$ and is able to evaluate the degree of inductive bias in terms of a ratio of positive accuracy and negative accuracy [He and Garcia, 2009].

$$G\text{-}score = \sqrt{Precision \cdot Recall}$$

As both $F\text{-}score$ and $G\text{-}score$ concentrate their measures on one class (positive examples) [Sokolova *et al.*, 2006], to have a general way of comparing our test results, we altered the positive examples between the majority and minority classes when computing $F\text{-}score$ and $G\text{-}score$. Thus we

		Minority				Majority			
	AUC	**Precision**	**Recall**	**Fscore**	**Gscore**	**Precision**	**Recall**	**Fscore**	**Gscore**
b-kde									
Original	0.797	0.139	0.033	0.054	0.068	0.830	**0.995**	**0.905**	**0.909**
Dup	0.733	0.385	0.454	0.417	0.418	0.869	0.742	0.801	0.803
SMOTE	0.807	0.488	0.705	0.577	**0.587**	0.833	0.644	0.726	0.733
B-SMOTE	0.774	0.258	0.456	0.330	0.343	0.846	0.671	0.748	0.754
MWMOTE	0.794	0.396	**0.754**	0.520	0.547	0.836	0.557	0.669	0.682
ADASYN	0.802	0.395	0.632	0.487	0.500	0.817	0.598	0.691	0.699
RAMOboost	0.748	0.358	0.343	0.350	0.350	0.860	0.822	0.841	0.841
CGMOS	**0.842**	**0.536**	0.517	**0.526**	0.526	**0.908**	0.815	0.859	0.860
knn									
Original	0.821	**0.701**	0.521	0.598	0.604	0.902	**0.942**	**0.922**	**0.9217**
Dup	0.810	0.519	0.732	0.607	0.616	0.921	0.818	0.867	0.868
SMOTE	0.827	0.506	**0.804**	0.621	0.638	0.925	0.805	0.861	0.863
B-SMOTE	0.811	0.494	0.736	0.591	0.603	0.927	0.790	0.853	0.856
MWMOTE	0.832	0.504	0.792	0.616	0.632	**0.928**	0.794	0.856	0.858
ADASYN	0.825	0.495	0.786	0.607	0.623	**0.929**	0.786	0.851	0.854
RAMOboost	0.827	0.540	0.684	0.604	0.608	0.918	0.847	0.881	0.881
CGMOS	**0.840**	0.544	0.766	**0.636**	**0.646**	0.925	0.842	0.882	0.883
svm									
Original	0.792	**0.632**	0.587	0.609	0.609	0.882	0.935	0.908	0.908
Dup	0.815	0.543	0.436	0.484	0.487	**0.981**	0.861	0.917	0.919
SMOTE	0.844	0.579	0.726	0.644	0.648	0.879	0.844	0.861	0.861
B-SMOTE	0.832	0.475	0.729	0.575	0.588	0.893	**0.959**	**0.924**	**0.925**
MWMOTE	0.830	0.547	0.647	0.593	0.595	0.880	0.884	0.882	0.882
ADASYN	0.827	0.536	0.654	0.589	0.592	0.880	0.755	0.813	0.815
RAMOboost	0.842	0.556	0.673	0.609	0.611	0.968	0.852	0.906	0.908
CGMOS	**0.864**	0.555	**0.788**	**0.651**	**0.661**	0.943	0.830	0.883	0.885
nn									
Original	0.801	**0.632**	0.412	0.499	0.510	0.892	**0.962**	**0.925**	**0.9258**
Dup	0.843	0.543	0.777	0.639	0.650	0.926	0.819	0.869	0.871
SMOTE	0.840	0.555	0.750	0.638	0.645	0.921	0.820	0.868	0.869
B-SMOTE	0.841	0.475	0.779	0.590	0.608	0.924	0.802	0.859	0.861
MWMOTE	0.841	0.547	0.778	0.642	0.652	0.927	0.812	0.866	0.867
ADASYN	0.842	0.536	**0.786**	0.637	0.649	0.929	0.803	0.861	0.863
RAMOboost	0.841	0.556	0.743	0.636	0.643	0.919	0.830	0.872	0.873
CGMOS	**0.865**	0.579	0.750	**0.653**	**0.659**	**0.933**	0.845	0.887	0.888
rf									
Original	0.872	**0.699**	0.534	0.606	0.611	0.909	**0.956**	**0.932**	**0.932**
Dup	0.873	0.682	0.641	0.661	0.661	0.917	0.924	0.921	0.921
SMOTE	0.875	0.667	0.655	0.661	0.661	0.920	0.917	0.918	0.918
B-SMOTE	0.867	0.653	0.637	0.645	0.645	0.920	0.906	0.913	0.913
MWMOTE	0.878	0.658	0.651	0.655	0.655	0.920	0.922	0.921	0.921
ADASYN	0.876	0.663	0.669	0.666	0.666	0.919	0.915	0.917	0.917
RAMOboost	0.874	0.686	0.618	0.650	0.651	0.915	0.933	0.924	0.924
CGMOS	**0.884**	0.685	**0.678**	**0.681**	**0.681**	**0.923**	0.926	0.924	0.924
Adaboost.M1									
Original	0.868	**0.699**	0.572	0.629	0.632	0.906	**0.944**	**0.925**	**0.9247**
Dup	0.865	0.622	0.708	0.662	0.664	0.922	0.873	0.897	0.897
SMOTE	0.867	0.608	0.714	0.657	0.659	0.923	0.880	0.901	0.901
B-SMOTE	0.864	0.581	0.724	0.644	0.648	**0.927**	0.861	0.893	0.893
MWMOTE	0.868	0.600	0.708	0.650	0.652	0.922	0.880	0.901	0.901
ADASYN	0.867	0.599	0.726	0.657	0.660	0.925	0.873	0.898	0.899
RAMOboost	0.865	0.631	0.699	0.663	0.664	0.922	0.882	0.901	0.902
CGMOS	**0.871**	0.619	**0.728**	**0.670**	**0.672**	0.925	0.882	0.903	0.903

Table 2: A summary of AUC, *Precision*, *Recall*, *F-score* and *G-score* of all competitors for the majority and minority classes produced by 6 classifiers on the artificial datasets.

	CGMOS	**Original**	**Dup**	**SMOTE**	**B-SMOTE**	**MWMOTE**	**ADASYN**	**RAMOboost**
BankMarket	**0.728**	0.661	0.708	0.718	0.710	0.721	0.710	0.723
BloodService	**0.733**	0.653	0.648	0.649	0.651	0.720	0.714	0.728
BreastCancer	0.992	0.992	**0.993**	0.992	0.989	0.991	0.991	0.992
BreastTissue	**0.984**	0.899	0.946	0.932	0.917	0.937	0.908	0.943
CarEvaluation	**0.997**	0.995	0.845	**0.997**	0.994	0.996	**0.997**	0.995
Card'graphy	**0.977**	0.976	0.939	0.962	0.956	0.925	0.957	0.960
CharacterTraj	0.985	0.962	0.717	0.985	0.978	0.981	**0.988**	0.909
Chess	**0.977**	0.974	0.959	0.973	**0.977**	0.974	0.975	0.959
ClimateSim	**0.908**	**0.908**	0.861	0.902	0.863	0.901	0.901	0.882
Contraceptive	**0.724**	0.705	0.699	0.712	0.702	0.705	0.702	0.705
Fertility	**0.673**	0.615	0.594	0.634	0.592	0.604	0.639	0.638
Haberman	**0.651**	0.623	0.577	0.600	0.593	0.594	0.587	0.586
ILPD	0.707	0.687	0.693	**0.715**	0.703	0.702	0.693	0.703
ImgSeg	**0.999**	0.998	**0.999**	0.997	0.998	0.998	0.997	0.998
Leaf	**0.908**	0.880	0.782	0.852	0.775	0.836	0.839	0.821
Libras	**0.945**	0.922	0.859	0.929	0.886	0.936	0.923	0.883
MultipleFs	**0.998**	**0.998**	0.997	**0.998**	0.997	0.997	0.996	0.997
Parkinson	0.841	0.676	0.692	0.834	0.791	0.837	**0.842**	0.760
PlanRelax	0.472	0.457	0.494	0.469	0.445	0.467	**0.488**	0.464
QSAR	**0.901**	0.886	0.879	0.895	0.863	0.886	0.886	0.882
SPECT	**0.820**	0.772	0.803	0.808	0.811	0.752	0.801	0.799
SPECTF	0.819	0.819	0.800	0.805	0.816	0.812	**0.825**	0.795
SeismicBumps	**0.743**	0.735	0.712	0.727	0.740	0.732	0.715	0.691
Statlog	**0.998**	0.992	0.996	**0.998**	0.990	0.996	0.976	0.996
PlatesFaults	**0.956**	0.928	0.844	0.954	0.920	0.943	**0.956**	0.881
TAEvaluation	**0.748**	0.682	0.644	0.703	0.671	0.707	0.665	0.657
UserKnowledge	**0.958**	0.837	0.919	0.953	0.947	0.951	0.950	0.888
Vertebral	**0.890**	0.839	0.869	0.855	0.829	0.860	0.794	0.872
Customers	**0.952**	0.930	0.943	0.946	0.884	0.902	0.946	**0.952**
Yeast	**0.925**	0.792	0.844	0.907	0.898	0.900	0.906	0.851
Average	**0.864**	0.827	0.808	0.844	0.830	0.842	0.842	0.830

Table 3: A summary of AUC of 8 oversampling algorithms over all 30 datasets used in our evaluation. The AUC is averaged over all 6 base classifiers used in the evaluation. It could be seen from above table that CGMOS achieves best AUC measures for 24 datasets out of 30. By average, the AUC of CGMOS is at least 2 percent higher than all other competitors.

show $F\text{-}score$ and $G\text{-}score$ for the majority and the minority classes separately.

Although, both $F\text{-}score$ and $G\text{-}score$ are great evaluation metrics, they are still less effective in some situations. So we also employ the ROC graphs [Fawcett, 2004][Fawcett, 2006][Mohri, 2005] in the evaluation. ROC graph is a two-dimensional graph, while $FP\ rate$ and $TP\ rate$ are its X axis and Y axis respectively.

An ROC graph basically manifest its usefulness by showing relative trade-off between benefit (true positive) and cost (false positive). One attractive property make ROC graph a good metric in imbalanced learning lies in the facts that ROC curve is insensitive to changes in class distribution. Because of this property, it is easier to see the performances of models trained by dataset oversampled by different algorithms. The goal in ROC space is to let curves be as close to upper-left-hand corner as possible, in which case the ratio between benefit and cost is maximized. To compare all test results in a more straightforward way, we also compute area under an ROC curve (AUC) which reduce the ROC performance to a single scalar value representing expected performance of the ROC curve.

4.5 Results

This section presents the performance of CGMOS and all the other methods on 30 real-world datasets. The same experiment procedure as the one in the experiments of the artificial dataset was conducted. All results are averged from 10 rounds of 10-folds cross-validations. A summary of the experiment results is shown in Table 2 and ROC graphs are shown in Figure 2.

Considering the classification results of the minority class, it can be observed that the proposed approach outperforms most of the compared methods under all classification algorithms in terms of $F\text{-}score$ and $G\text{-}score$. For $F\text{-}score$ and $G\text{-}score$ of the majority class, the proposed approach in most cases is only second to the original data without oversampling. This is because the original dataset is imbalanced and

	Knn	**Rf**	**B-kde**	**Nn**	**Svm**	**Boost**
Original	5e-5	1e-4	0.004	1e-4	0.026	0.04
Dup	2e-6	5e-5	3e-6	0.03	0.049	0.004
SMOTE	0.003	2e-4	6e-6	0.018	0.006	0.046
B-SMOTE	4e-6	7e-6	2e-5	5e-4	0.047	5e-4
MWMOTE	0.046	4e-5	1e-5	0.003	0.005	0.007
ADASYN	8e-6	7e-5	9e-5	0.005	1e-4	0.003
RAMOboost	2e-6	5e-5	3e-6	0.001	0.045	0.035

Table 4: A summary of p-values of statistical significant tests of classification results using CGMOS against each of all the other competitors.

it favors the majority class more than the minority class during classification. Overall, CGMOS achieves the best AUC over all tests. This is because the proposed approach takes into account both of the majority and minority classes and increases the certainties of the two classes while oversampling.

The same conclusion can be made from the ROC curves shown in Fig. 2. It could be seen from the ROC curves that the proposed approach has the highest values almost everywhere. The proposed approach achieves the best result when random forest is used as the classifier. For b-kde as the classifier, the proposed approach gets the largest improvement since the design of the proposed approach uses b-kde for certainty computations.

To get a closer view of the performances of all compared methods on each dataset, we show the AUC results of CGMOS and all other compared methods for each dataset in Table 3. The table shows that by average the AUC of CGMOS is 2 percent higher than SMOTE whose AUC is 2nd highest.

Previous studies show that it is not necessary for a learning procedure to obtain best classification results when a dataset is perfectly balanced[Batista *et al.*, 2004][Weiss and Provost, 2003]. How much to oversample is usually empirically determined [Chawla *et al.*, 2004]. To evaluate this aspect we performed another experiment in which we synthesized increasing number of minority samples and investigated how different amounts of new samples impact classification results.

Let δ denote the difference of data samples between the majority and the minority class. We performed multiple experiments where in each round we synthesized $k\delta$ new samples of the minority class where k gradually increased from 0.5 to 5. The classification results are shown in Figure 3. As can be observed in the results, CGMOS achieves the best results in all cases. Also, observe that when increasing the number of data samples added, the results of CGMOS are much more robust compared with other approaches. Note that the results of some methods such as Dup(b-kde), B-SMOTE(knn) and B-SMOTE(Adaboost.M1) are even lower than the results at the starting point where datasets are not oversampled. This highlights the advantage of CGMOS when handling oversampling on boundary samples.

4.6 Statistical Significance Analysis

We evaluate the statistical significance of the classification results of all competitors. Statistical significance plays a critical role in determining whether a null hypothesis should be rejected or retained, where the term null hypothesis refers to a general statement that sample observations result purely from chance. For a null hypothesis to be rejected as false, the result has to be identified as being statistically significant.

To determine whether to reject a null hypothesis, a p-value has to be calculated, which is the probability of observing an effect given that the null hypothesis is true [Devore, 2011]. The null hypothesis is rejected if p-value is less than the significance level. The significance level is the probability of rejecting the null hypothesis given that it is true. The lower the significance level the more confident we can be in replicating the results and usually the significance level is set at 5%. Then a sample observation is determined to be statistically significant if p-value is less than 5%, which is formally written as $p < 0.05$ [McKillup, 2006].

We follow the same protocols used in [Demšar, 2006][Chen *et al.*, 2010][Barua *et al.*, 2014] and choose to use Wilcoxon signed-ranks test in this paper. Wilcoxon signed-ranks test is a nonparametric statistical procedure for comparing two samples that are paired, or related [Corder and Foreman, 2009]. Different from t-test [PhD, 2008][Zimmerman, 1997][Demšar, 2006] whose null hypothesis is that the mean difference between pairs is zero, the null hypothesis of Wilcoxon signed-ranks test is that the median difference between pairs of observations is zero.

The test results are shown in Table 4. It could be seen from the table that the p-value of all tests are smaller than 0.05 and pass the test.

5 Conclusion

In this paper, we address the imbalanced binary classification problem by proposing a novel minority oversampling strategy. Different from existing approaches, CGMOS does not randomly synthesize new data along decision boundaries. Instead, CGMOS computes the Bayes classification certainties for both the majority and minority classes and then synthesize new samples based on improvement of the certainties for samples in both classes. We prove that CGMOS can achieve better classification results compared with SMOTE. In addition, experimental results show that CGMOS outperforms known oversampling techniques using various metrics.

References

[Barua *et al.*, 2011] Sukarna Barua, Md Monirul Islam, and Kazuyuki Murase. A novel synthetic minority oversampling technique for imbalanced data set learning. In *Neural Information Processing*, pages 735–744. Springer, 2011.

[Barua *et al.*, 2014] Simul Barua, Md Minarul Islam, Xin Yao, and Kazuyuki Murase. Mwmote–majority weighted minority oversampling technique for imbalanced data set learning. *Knowledge and Data Engineering, IEEE Transactions on*, 26(2):405–425, 2014.

[Batista *et al.*, 2004] Gustavo EAPA 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*, 6(1):20–29, 2004.

[Chawla *et al.*, 2002] 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.

[Chawla *et al.*, 2003] Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. Smoteboost: Improving prediction of the minority class in boosting. *Subseries of Lecture Notes in Computer Science*, page 107, 2003.

[Chawla *et al.*, 2004] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial: special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, 6(1):1–6, 2004.

[Chen *et al.*, 2010] Sheng Chen, Haibo He, Edwardo Garcia, et al. Ramoboost: Ranked minority oversampling in boosting. *Neural Networks, IEEE Transactions on*, 21(10):1624–1642, 2010.

[Corder and Foreman, 2009] Gregory W. Corder and Dale I. Foreman. *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach*. Wiley, 2009.

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

[Devore, 2011] Jay L. Devore. *Probability and Statistics for Engineering and the Sciences*. Duxbury Press, 2011.

[Drummond and Holte, 2003] Chris Drummond and Robert C. Holte. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Workshop on Learning from Imbalanced Data Sets II, International Conference on Machine Learning*, pages 1–8, 2003.

[Elgammal *et al.*, 2002] Ahmed Elgammal, Ramani Duraiswami, David Harwood, and Larry S Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, 2002.

[Fawcett, 2004] Tom Fawcett. Roc graphs: Notes and practical considerations for researchers. *Machine learning*, pages 1–38, 2004.

[Fawcett, 2006] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[Freund and Schapire, 1996] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In Lorenza Saitta, editor, *Proceedings of the Thirteenth International Conference on Machine Learning (ICML 1996)*, pages 148–156. Morgan Kaufmann, 1996.

[Guo and Viktor, 2004] Hongyu Guo and Herna L Viktor. Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *ACM SIGKDD Explorations Newsletter*, 6(1):30–39, 2004.

[Han *et al.*, 2005] 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*, pages 878–887. Springer, 2005.

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

[He *et al.*, 2008] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1322–1328. IEEE, 2008.

[Lichman, 2013] M. Lichman. UCI machine learning repository, 2013.

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

[McKillup, 2006] Steve McKillup. *Statistics Explained: An Introductory Guide for Life Scientists*. Cambridge University Press, 2006.

[Mohri, 2005] C Mohri. Confidence intervals for the area under the roc curve. In *Advances in neural information processing systems*, page 305, 2005.

[PhD, 2008] Barbara Fadem PhD. *High-Yield(TM) Behavioral Science (High-Yield Series)*. LWW, 2008.

[Quinlan, 1993] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[Sharma and Bilgic, 2013] Manali Sharma and Mustafa Bilgic. Most-surely vs. least-surely uncertain. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 667–676, 2013.

[Sokolova *et al.*, 2006] Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *AI 2006: Advances in Artificial Intelligence*, pages 1015–1021. Springer, 2006.

[Weiss and Provost, 2003] Gary M Weiss and Foster Provost. Learning when training data are costly: the effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, pages 315–354, 2003.

[Weiss, 2004] Gary M Weiss. Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter*, 6(1):7–19, 2004.

[Zhang and Mani, 2003] Jianping Zhang and Inderjeet Mani. Knn approach to unbalanced data distributions: A case study involving information extraction. In *Int'l Conf. Machine learning, workshop learning from imbalanced data sets*, 2003.

[Zhang *et al.*, 2006] Xibin Zhang, Maxwell L King, and Rob J Hyndman. A bayesian approach to bandwidth selection for multivariate kernel density estimation. *Computational Statistics & Data Analysis*, 50(11):3009–3031, 2006.

[Zimmerman, 1997] Donald W Zimmerman. Teacher's corner: A note on interpretation of the paired-samples t test. *Journal of Educational and BEhavioral Statistics*, 22(3):349–360, 1997.