Synthetic Information towards Maximum Posterior Ratio for deep learning on Imbalanced Data: SIMPOR

Hung Nguyen[a], Morris Chang[a]

[a] *University of South Florida, USA*

Abstract

Most state-of-the-art machine learning and deep learning classification techniques assume input data are class-balanced. In fact, it is common in real-world applications that some classes naturally contain significantly less data than others. This reduced ML algorithms' performance of classifiers because they are biased toward the majority class. While there have been a number of solutions proposed for conventional machine learning algorithms (e.g., SVM, regression family), there is a lack of research about imbalanced data for deep learning models.

This work explores how imbalanced data affects deep learning and proposes a data balancing technique by generating synthetic data in the minority class. Instead of generating data randomly, our approach prioritizes balancing the most informative minority samples. Moreover, we maximizes the probability that generated synthetic data fall into the minority class. Elaborately, not all samples contribute equally to the models; only the ones located in the informative region carry significant information. Thus, we start with finding and balancing such informative instances by leveraging an entropy-based active learning technique, which results in high entropy samples. Since these are practically near class boundaries and might be disputed by different classes, generating synthetic data in this region is critical and can accidentally break data topology. Therefore, we safely generate surrounding neighbors of each minority sample to preserve data topology. More importantly, we ensure that the synthetic samples fall into the targeted class (minority class) and fall apart from the majority class by maximizing the posterior ratio between classes derived from Bayes' Theorem. Experimental results show that our technique constantly outperforms widely-used techniques over different settings of imbalance ratio and data dimension.

Keywords: data imbalance, deep learning, maximum fractional posterior, informative samples

1. Introduction

Data imbalance is a common phenomenon; it could be caused by sampling procedures or simply the nature of data. For example, it is difficult to sample some of the rare diseases in the medical field, so collected data for these are usually significantly less than that for other diseases. This leads to the problem of class imbalance in machine learning. The chance of rare samples appearing in model training processes is much smaller than that of common samples. Thus, machine learning models will be dominated by the majority class; this results in a higher prediction error rate. Existing work also observed that imbalanced data cause a slow convergence in the training process because of the domination of gradient vectors coming from the majority class [32, 22]. In the last decades, a number of techniques have been proposed to soften the negative effects of data imbalance on conventional machine learning algorithms by analytically studying particular algorithms and developing corresponding strategies. However, the problem for heuristic algorithms such as deep learning is often more difficult to tackle. In this work, we address data imbalance for deep learning models by providing a solution that utilizes both deep active learning techniques and statistical derivations of Bayes' Theorem.

We categorize existing solutions into model-centric and data-centric approaches in which the first approach aims at modifying machine algorithms, and the latter looks for data balancing techniques, respectively. Perhaps data-centric methods are more commonly used because they do not tie to a specific model. In this category, a simple data balancing technique is to duplicate minority instances to balance the sample quantity between classes. This can preserve the best data structure and reduce the negative impact of data imbalance to some degree. However, this puts too much weight on a very few minority samples; as a result, it causes over-fitting problems in deep learning when the imbalance ratio becomes higher.

Another widely-used method in this category is Synthetic Minority Oversampling Technique (SMOTE) [4], which randomly generates synthetic data on the connections (in Euclidean space) between minority samples. However, this easily breaks data topology, especially in high-dimensional space, because it can accidentally connect instances that are not supposed to be connected. In addition, if there are minority samples located in the majority class, the method will generate sample lines across the decision boundary, which leads to distorted decision boundaries and misclassification. To improve SMOTE, Hui Han, *et al.* [16] proposed a SMOTE-

Email addresses: nsh@usf.edu (Hung Nguyen), chang5@usf.edu (Morris Chang)

based method (Borderline SMOTE), in which they only apply SMOTE on the near-boundary samples determined by the labels of their neighbors. For example, if a sample Euclidean space-based group includes samples from other classes, they can be considered as samples near the border. Since this method is entirely based on Euclidean distance from determining neighbors to generating synthetic data, it performs poorly in high dimensional space. Similar to SMOTE, if there is any misgenerated sample near the boundary, it will worsen the problem due to synthetic samples bridges across the bounder. Leveraging the same way as SMOTE generates synthetic samples, another widely-used technique, ADASYN [17], controls the number of generated samples by the number of samples in different classes within small groups. Again, this technique still suffers distortion of the decision boundary in the case the boundary region is imbalanced.

To alleviate the negative effects of data imbalance and avoid the drawbacks of existing techniques, we propose a minority oversampling technique that focuses on balancing at the informative region that provides the most important information to the deep learning models. Besides, the technique enhances the chance that synthetic data fall into the minority class so that it will not cause more errors to the model. By carefully generating synthetic data near minority samples, our proposed technique also preserves the best data topology.

To find informative samples, we leverage an entropy-based deep active learning technique that is able to select samples yielding high entropy to deep learning models. We denote where the informative samples are located as the informative region. We then balance this region first and the remaining data are balanced later so that it would reduce the decision distortion mentioned earlier. For each minority sample in this region, we safely generate its synthetic neighbors so that the global data topology is still preserved. However, generating synthetic samples in this region is critical because it can easily fall across the decision boundary. Therefore, we design a direction to generate synthetic samples that maximize their posterior probability of belonging to the minority class based on Bayes's Theorem. However, maximizing the posterior probability is facing infeasible computation in the denominator. To overcome this, we maximize the posterior ratio instead, so that the denominator will disappear. This also ensures that the synthetic samples are not only close to the minority class but also far from the majority class. The remaining data are eventually balanced by randomly generating neighbors for each sample.

The proposed technique results in a balanced dataset that improves the training performance and alleviates the class imbalance problem. Our experiments indicate that we can achieve better classification results over widely-used techniques in all experimental cases by applying the proposed strategy.

Our work has the following main contributions:

1. Exploring the impact of class imbalance on deep learning.
2. proposing a minority oversampling technique, namely Synthetic Information towards Maximum Posterior Ratio, to balance data classes and alleviate data imbalance impacts. Our technique is enhanced by following key points.
   (a) Leveraging an entropy-based active learning technique to prioritize the region that needs to be balanced. It is the informative region where samples provide high information entropy to the model.
   (b) Leveraging Maximum Posterior Ratio and Bayes's theorem to determine the direction to generate synthetic minority samples to ensure the synthetic data fall into the minority class and not fall across the decision boundary.
   (c) Approximating the likelihood in the posterior ratio using kernel density estimation, which can approximate a complicated statistical model. Thus, the proposed technique is able to work with large, distributively complex data.
   (d) Carefully generating synthetic samples surrounding minority samples so that the global data topology is still preserved.

The rest of this paper is organized as follows. Section 2 introduces related concepts that will be used in this work, i.e., Imbalance Ratio, Macro F1-score, and Entropy-based active learning. Section 3 will provide more detail on the problem of learning from an imbalanced dataset. Our proposed solution to balance dataset, Synthetic Information towards Maximum Posterior Ratio, will be explained comprehensively in Section 4. Section 5 discusses the technique's implementation and complexity. We will show experiments on different datasets, including artificial and real datasets in Section 6. We also discuss experimental results in the same section. In Section 7, we briefly review other existing works. Section 8 concludes the work and discusses future work.

2. Preliminaries

In this section, we introduce related concepts that will be utilized in our work.

2.1. Imbalance Ratio (IR)

For binary classification, we use imbalance ratio (IR) to depict the data imbalance as it has been widely used. IR is the ratio of the majority class samples to the minority class's samples. For example, if a dataset contains 1000 class-A samples and 100 class-B samples, the Imbalance Ratio is 10:1.

2.2. F1 Score

In this work, we evaluate balancing data techniques by the classification results on balanced data. To measure the accuracy of classification, we use Macro-averaging F1-Score, in which we compute F1 scores per class and average with the same weight regardless of how often they appear in the dataset. The F1 score is computed based on two factors Recall and Precision as follows:

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$F1 - score = \frac{2 * Recall * Precision}{Recall + Precision}, \tag{3}$$

where T and F stand for True and False; P and N stand for Positive and Negative.

2.3. Entropy-based Active Learning

To find informative samples, we leverage entropy-based active learning. The method gradually selects batch-by-batch samples that provide high information to the model based on information entropy theory [30]. The information entropy is quantified based on the "surprise" to the model in terms of class prediction probability. Take a binary classification for example, if a sample is predicted to be 50% belonging to class A and 50% belonging to class B, this sample has high entropy and is informative to the model. In contrast, if it is predicted to be 100% belonging to class A, it is certain and gives zero information to the model. The class entropy E for each sample can be computed as follows.

$$E(x, \theta) = -\sum_{i}^{n} P_\theta(y = c_i|x) \log_n P_\theta(y = c_i|x) \tag{4}$$

where $P_\theta(y = c_i|x)$ is the probability of data x belonging to the ith class of n classes with current model parameter θ.

In this work, we consider a dataset containing N pairs of samples X and corresponding labels y, and a deep neural network with parameter θ. At the first step $t^{(0)}$, we train the classifier with parameter $\theta^{(0)}$ on a random batch of k labeled samples and use the $\theta^{(0)}$ to predict the labels for the rest of the data (we assume their labels are unknown). We then compute the prediction entropy of each sample based on Equation 4. We are now able to collect the first batch of informative samples by selecting k samples based on the top k highest entropy. We query labels for this batch and concatenate to existing labeled data to train the classifier parameter $\theta^{(1)}$ in the next step $t^{(1)}$. Steps are repeated until all sample's entropy are less than a threshold e.g., $Threshold = 0.7$.

3. The Problem of Learning From Imbalanced Datasets

In this section, we review the problem of learning from imbalanced datasets. Although the problem may apply to different machine learning methods, we focus on deep learning in this work.

Figure 1 illustrates our problem on a binary classification. The imbalance in the informative region (light blue eclipse) could lead to separation errors. The dashed green line depicts the expected boundary, while the solid blue line is the model's boundary. Since the minority class is lacking data in this region, the majority class will dominate the model even with a few noisy samples, and this leads to a shift of the model's boundary.

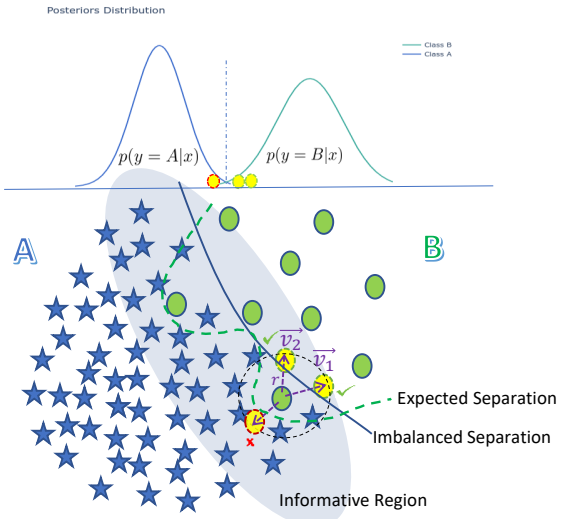


Figure 1: Learning from imbalanced datasets

In contrast to the study by Ertekin *et al.* [8] which assumes the informative region is more balanced by nature and proposes a solution that only classifies over the informative samples, our assumption is different. We consider the case that the informative region contains high imbalanced data, which we believe happens in most of the real scenarios. In a more complex setting such as high dimensional and topologically complex data, the problem could be more severe. Therefore, we proposed a technique to tackle the problem of data imbalance by oversampling the minority class in an informative manner. The detail of our proposed technique will be described in Section 4.

4. Synthetic Information towards Maximum Posterior Ratio

To alleviate the negative effects of data imbalance, we propose a comprehensive approach, Synthetic Information towards Maximum Posterior Ratio (SIMPOR), which aims to generate synthetic samples for minority classes. We first find the informative region where informative samples are located and balance this region by creating synthetic surrounding neighbors for minority samples. The remaining region is then fully balanced by arbitrarily generating minority samples' neighbors. We elaborate on how our strategy is developed in the rest of this section.

4.1. Methodology Motivation

As Chazal and Michel mentioned in their work [19], the natural way to highlight the global topological structure of the data is to connect data points' neighbors; our proposed method preserves their observation but in the reverse procedure, generating surrounding synthetic neighbors for minority samples. Thus, our method not only generates more data for minority class but also preserve the underlying topological structure of the entire data.

Agreeing with the mutual idea in [8] and [3], we believe that informative samples play the most important role in the

prediction success of both traditional machine learning models (e.g., SVM, Naive Bayes) and modern deep learning approaches (e.g., neural network). Thus, our method finds these informative samples and focuses on augmenting minority data in this region. In this work, we apply an entropy-based active learning strategy mentioned in 2.3 to find the samples that maximize entropy to the model. This strategy is perhaps the most popular active learning technique and over-performs many other techniques on several datasets [12], [26] [29].

4.2. Generating minority synthetic data

A synthetic neighbor x' and its label y' can be created surrounding a minority sample x by adding a small random vector v to the sample, $x' = x + v$. This lays on the d-sphere surface centered by x, and the d-sphere's radius is set by the length of vector $\vec{v}$, $|\vec{v}|$. It is, however, critical to generate synthetic data in the informative region because synthetic samples can unexpectedly jump across the decision boundary. This can be harmful to the model as this might create outliers and reduce the model's performance. Therefore, we safely find vector $\vec{v}$ towards the minority class such as $\vec{v}_0$ and $\vec{v}_1$ depicted in Figure 1. Our technique is described via a binary classification scenario as follows.

Let consider a binary classification problem between majority class A and minority class B. From the Bayes' theorem, the posterior probabilities $p(y' = A|x')$ or $p(y' = B|x')$ can be used to present the probabilities that a synthetic sample x' belongs to class A or class B, respectively. Let the two posterior probabilities be f_0 and f_1; they can be expressed as follows.

$$p(y' = A|x') = \frac{p(x'|y' = A)\ p(A)}{p(x')} = f_0 \tag{5}$$

$$p(y' = B|x') = \frac{p(x'|y' = B)\ p(B)}{p(x')} = f_1 \tag{6}$$

As mentioned earlier, we would like to generate each synthetic data x' that maximizes the probability of x' belonging to the minority class B and minimizes the chance x' falling into the majority class A. Thus, we propose a method that maximizes the fractional posterior f,

$$f = f_1/f_0 \tag{7}$$

$$= \frac{p(x'|y' = B)\ p(B)}{p(x'|y' = A)\ p(A)}. \tag{8}$$

Approximation of likelihoods in Equation 8: We use non-parametric kernel density estimates (KDE) to approximate the likelihoods $p(x'|y' = A)$ and $p(x'|y' = B)$ as KDE is flexible and does not require specific assumptions about the data distribution. One can use a parametric statistical model such as Gaussian to approximate the likelihood; however, it oversimplifies the data and does not work effectively with topological complex data, especially in high dimensions. In addition, parametric models require an assumption about the distribution of data which is difficult in real-world problems since we usually do not have such information. On the other hand, KDE only needs a kernel working as a window sliding through the data. Among different commonly used kernels for KDE, we choose Gaussian Kernel as it is a powerful continuous kernel that would also eases the derivative computations for finding optima.

Approximation of priors in Equation 8: Additionally, we assume prior probabilities of observing samples in class A ($p(A)$) and class B ($p(B)$) (in Equation 8) are constant. Hence, these probabilities do not affect our algorithm in terms of generating synthetic neighbors for minority samples because we only determine the relative direction between the minority and the majority class. Thus, they can be canceled out at the end of the equation reduction.

Equation 8 reduction: Let X_A and X_B be the subsets of dataset X which contain samples of class A and class B, $X_A = \{x : y = A\}$ and $X_B = \{x : y = B\}$. N_A and N_B are the numbers of samples in X_A and X_B. d is the number of data dimensions. h presents the width parameter of the Gaussian kernel. The posterior ratio for each synthetic sample x' then can be estimated as follows:

$$f = \frac{p(x'|y' = B)\ p(B)}{p(x'|y' = A)\ p(A)} \tag{9}$$

$$\propto \frac{\frac{1}{N_B h^d}\ \sum_{i=1}^{N_B} (2\pi)^{-\frac{d}{2}}\ e^{\frac{1}{2}(\frac{x' - X_{B_i}}{h})^2}\ p(B)}{\frac{1}{N_A h^d}\ \sum_{j=1}^{N_A} (2\pi)^{-\frac{d}{2}}\ e^{\frac{1}{2}(\frac{x - X_{A_j}}{h})^2}\ p(A)} \tag{10}$$

$$\propto \frac{N_A}{N_B}\ \frac{\sum_{i=1}^{N_B} e^{\frac{1}{2}(\frac{x' - X_{B_i}}{h})^2}\ p(B)}{\sum_{j=1}^{N_A} e^{\frac{1}{2}(\frac{x' - X_{A_j}}{h})^2}\ p(A)} \tag{11}$$

$$\propto \frac{\sum_{i=1}^{N_B} e^{\frac{1}{2}(\frac{x' - X_{B_i}}{h})^2}}{\sum_{j=1}^{N_A} e^{\frac{1}{2}(\frac{x' - X_{A_j}}{h})^2}}. \tag{12}$$

Finding synthetic samples surrounding a minority sample: Because we want to generate neighbors for each minority sample that maximizes Function f in Equation 12, we examine points lying on the sphere centered at the minority sample with a small radius r. As a result, we can find a vector $\vec{v}$ so that it can be added to the sample to generate a new sample. The relationship between a synthetic sample x' and a minority sample can be described as follows,

$$\vec{x'} = \vec{x} + \vec{v}, \tag{13}$$

where $|\vec{v}| = r$, and r is sampled from a uniform distribution $r \sim U(0, R)$, $0 < r < R$. The range parameter R is relatively small and computed as the average distance of k-nearest neighbors of the minority sample x to itself. This will ensure that the generated sample will be surrounding the minority sample. Consider a minority sample x and its k-nearest neighbors in the Euclidean space, R can be computed as follows:

$$R = \frac{1}{k} \sum_{1}^{k} \|x - x_j\|, \tag{14}$$

where $\|x - x_j\|$ is the Euclidean distance between a minority sample x and its jth neighbor. k is a parameter indicating number of the neighbors. In practice, we prefer to tune k from 5 to 20.

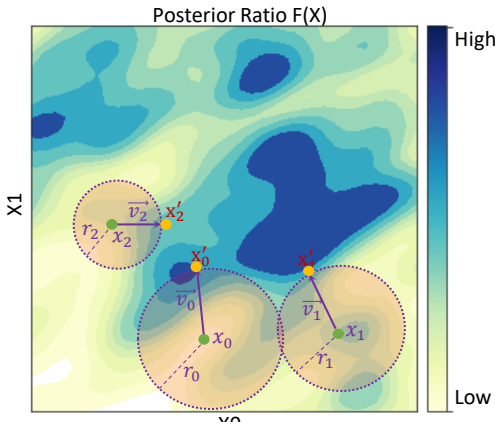


Figure 2: Demonstration on how SIMPOR generates three synthetic samples x'_0, x'_1, x'_2, from three minority samples x_0, x_1, x_2, by maximizing the Posterior Ratio.

Figure 2 depicts a demonstration of finding 3 synthetic samples from 3 minority samples. In fact, one minority can be re-sampled to generate more than one synthetic sample. For a minority sample x_0, we find a synthetic sample x'_0 by maximizing the objective function $f(x'_0), x'_0 \in X$ with a constraint that the Euclidean length of $\vec{v_0}$ equals to a radius r_0, $\|\vec{v_0}\| = r_0$ or $\|\vec{x'_0} - \vec{x_0}\| = r_0$ (derived from Equation 13).

The problem can be described as a constrained optimization problem. For each minority sample x, we find a synthetic sample $x' \in \mathbb{R}^d$ lying on the d-sphere centered at x with radius r and maximizing function in Equation 12,

$$\max_{x'} f(x') \quad \text{s.t. } \|\vec{x'} - \vec{x}\| = r, \tag{15}$$

where $r \sim U(0, R)$.

Solving optimization problem in Equation 15: Interestingly, the problem in Equation15 is solvable. Function $f(x)$ in Equation 12 is defined and continuous for $x' \in (-\infty, +\infty)$ because all of the exponential components (Gaussian kernels) are continuous and greater than zero. In addition, the constraint, $\|\vec{x'} - \vec{x}\| = r$, which contains all points on the sphere centered at x with radius r is a closed set ([2]). Thus, a maximum exits as proved in [1]. To enhance the diversity of synthetic data, we accept either the global maximum or any local maximum, so that the synthetic samples will not simply go the same direction.

We solve the problem in Equation 15 by using the Projected Gradient Ascent approach in which we iteratively update the parameter to go up the gradient of the objective function. A local maximum is found if the objective value cannot be increased by any local update. For simplification, we rewrite the problem in Equation 15 by shifting the origin to the considered minority sample. The problem becomes finding the maximum of function $f(x')$, $x' \in \mathbb{R}^d$, constrained on a d-sphere, i.e., $\|x'\| = r$. Our solution can be described in Algorithm 1. After shifting the coordinates system, we start with sampling a random point on the constraint sphere (line $1-2$). The gradient of objective function at time t, $g_t(x'_t)$, is computed and projected onto sphere tangent plane as p_t (line $4-5$). It is then normalized and used for update a new x'_{t+1} by rotating a small angle $lr * \theta$ (line $6-7$). The algorithm stops when the value of $f(x')$ is not increased by any update of x'. We finally shift to the original coordinates and return the latest x'_t.

Algorithm 1 Sphere-Constrained Gradient Ascent for Finding Maximum

Input: A minority sample x_0, objective function $f(x, X)$
Parameter:
r : The radius of the sphere centered at x_0
θ : Sample space $\theta \in [0, 2\pi]$
lr : Gradient ascent learning rate
Output: An local maximum x'

1: Shift the Origin to x_0
2: Randomly initiate x'_t on the sphere with radius r
3: **while** converge condition **do**
4: Compute the gradient at x'_t
$g_t(x'_t) = \nabla f(x'_t)$
5: Project the gradient onto the sphere tangent plane
$p_t = g_t - (g_t \cdot x'_t)x_t$
6: Normalize projected vector
$p_t = p_t / \|p_t\|$
7: Update x' on the constrained sphere
$x'_{t+1} = x'_t cos(lr * \theta) + p_t sin(lr * \theta)$
8: **end while**
9: Shift back to the Origin
10: **return** x'_t

4.3. Algorithm

Our strategy can be described in Algorithm 2. Our algorithm takes an imbalanced dataset as its input and results in a balanced dataset which is a combination of the original dataset and synthetic samples. We first choose an active learning method $AL(\cdot)$ and find a subset of informative samples S, which is shown in lines $1-2$ in Algorithm 2. In this work, we choose entropy-based active learning for our experiments. We then generate synthetic data to balance S. For each random sample x_i^c in S and belonging to minority class c, we randomly sample a small radius r and find a synthetic sample that lies on the sphere centered at x_i^c and maximizes the posterior ratio in Equation 12 (lines $3-11$). The process is repeated until the informative set S is balanced. Similarly, the remaining region is balanced which can be described in the pseudo-code from line 12 to line 20. The final output of the algorithms is a balanced dataset D'.

5. Algorithm Implementation and Complexity

Our proposed method is straightforward in implementation. We first train a neural network model with initial samples and

Algorithm 2 SIMPOR

Input: Original Imbalance Dataset D including data X and labels y.
Parameter: MA is the majority class, MI is a set of other classes.
k: Number of neighbors of the considered sample which determines the maximum range of the sample to its synthetic samples.
$Count(c, P)$: A function to count class c sample number in population P.
$G(x_0, f, r)$: Algorithm 1, which returns a synthetic sample on sphere centered at x_0 with radius r and maximize Equation 12.
Output: Balanced Dataset D' including $\{X', y'\}$

Select an Active Learning Algorithm $AL()$
Query a subset of informative samples $S \in D$ using AL: $s \leftarrow AL(D)$
{Balance the informative region}
for $c \in MI$ **do**
  while $Count(c, S) \leq Count(MA, S)$ **do**
    Select a random $x_i^c \in S$
    Compute maximum range R based on k
    Randomly sample a radius $r \sim U(0, R)$
    Generate a synthetic neighbor x' from x_i^c: $x' = G(x_i^c, f, r)$
    Append x' to D'
  end while
end for
{Balance the remaining region}
for c in MI **do**
  while $Count(c, D') \leq Count(MA, D')$ **do**
    Select a random $x_j^c \in \{X - S\}$
    Compute maximum range R based on k
    Randomly sample a radius $r \sim U(0, R)$
    Generate a synthetic neighbor x' of x_j^c
    Append x' to D'
  end while
end for
return

start querying next batches data based on the entropy scores from previous model to find informative samples. The model is then updated with new batches of data until the entropy scores reach a certain threshold. All the informative samples are then balanced first, and the remaining data are balanced later. Each synthetic data point is generated by finding a local maxima in Equation 12.

Perhaps the costly part of SIMPOR is that each synthetic sample requires to compute a kernel density estimation of the entire dataset. Elaborately, let n be the number of samples of the dataset. In the worst case, the number of samples of minority and majority class are $N_B = 1$ and $N_A = n - 2$ respectively. We need to generate $n - 1$ synthetic samples to completely balance the dataset. Since each generated sample must loop through the entire dataset of size n to estimate the density, the complexity is $O(n^2)$.

Table 1: Classification models' setting for each dataset.

Model Setting	Moon	Breast Cancer	Credit Card
Hidden Layers	3	3	3
Neurons/Layer	100	50	200
Hidden Activation	ReLu	ReLu	ReLu
Output Activation	Softmax	Softmax	Softmax
Epochs	200	150	200
Batch size	32	32	64
Optimizer	Adam	Adam	Adam
Learning Rate	0.01	0.01	0.01

Although generating synthetic data is only a one-time process, and this does not affect the classification performance in the testing phase, we still alleviate its weakness by providing parallelized implementations. We provide two suggestions, multiple CPU thread-based and GPU-based implementations. While the former simply computes each synthetic data sample in a separated CPU thread, the later computes each exponential component in 12 parallelly in GPU's threads. More specifically, Equation 12 can be rewritten as N_B components of $e^{\frac{1}{2}(\frac{x-X_{B_i}}{h})^2}$ and N_A components of $e^{\frac{1}{2}(\frac{x-X_{A_i}}{h})^2}$. Fortunately, they are all independent and can be parallelly processed in GPUs. The latter is then implemented using Python Numba and Cupy libraries which utilize CUDA toolkit from NVIDIA [25]. The consumption time for kernel density estimation for each synthetic data point is then reduced by $N_A + N_B = n$ times, which significantly simplifies the complexity to $O(n)$. Our source code can be found on following Github link https://github.com/nsh135/SIMPOR.

6. Experiments and Discussion

In this section, we experiment on binary classification for both artificial dataset (i.e., Moon) for demonstration and real-world datasets (i.e., Breast Cancer, Credit Card Fraud). Samples in artificial Moon have two dimensions while samples in Breast Cancer and Credit Card Fraud are both 30-dimension numerical data. Compare to the original Breast Cancer dataset size (569 samples), the other original Credit Card dataset contains a much larger amount of data, 284,907 samples. The implementation steps to balance datasets are following Algorithm 2. To evaluate our proposed balancing technique, we compare the classification performance to different widely-used techniques. More specifically, We compare SIMPOR to SMOTE [4], Borderline-SMOTE [16], ADASYN [17], Random Oversampling, and Raw data which does not apply any balancing technique. To evaluate classifications performance for skewed datasets, we measure some powerful and widely-used metrics such as Recall, Precision, F1-score, Area Under The Curve (AUC).

6.1. Sharing Settings

This subsection describes settings sharing for all datasets. In order to find the informative subset, we leverage entropy-based

(a) Raw Data. F1 = 0.852

(b) SIMPOR. F1 = 0.917

(c) SMOTE. F1 = 0.862

(d) Random Over Sampling. F1 = 0.834

(e) BorderlineSMOTE. F1 = 0.827

(f) ADASYN. F1 = 0.817

Figure 3: Balanced training data plot, model's decision boundary plot and testing F1 score for Moon Dataset.

active learning as mentioned in Section 2.3. Classifier for active learning is a fully connected neural network model containing 3 hidden layers with *relu* activation functions and 100 neurons each layer. The output layer uses softmax activation function. The models are trained in a maximum of 300 epochs with the early stop option when the loss does not change after updating weights.

In addition, we randomly split the data into two parts, 80% for training and 20% for testing. Reported testing results for each dataset are the averages of 5 experimental trials. For SIMPOR to find optima of the function in Equation 15, we use a gradient ascent rate of 0.00001 and the maximum iteration of 300. The architecture detail of the evaluation model for each dataset is described in Table 1.

6.2. SIMPOR on artificial Moon dataset

We implement our technique on an artificial 2-dimension numeric dataset as a demonstration of our proposed method. Figure 3 captures the classification F1 results for different techniques. We also visualize model decision boundaries to provide additional information on how the classification models are affected. To classify the data, we use a fully connected neural network which is described in Table 1.

6.2.1. Dataset

We first generate a balanced dataset using python library *sklearn.datasets.make_moons* including 3000 two-dimensional samples labeled in two classes, A and B. We then create an imbalanced dataset with an Imbalance Ratio of 3:1 by randomly removing 1000 samples from class B. As a result, the training dataset becomes imbalanced as visualized in Figure 3a, which contains 400 samples of class B and 1200 samples of class A. The testing data contains 100 samples of class B and 300 samples of class A.

6.2.2. Results and Discussion

From the results shown in Figure 3, it is clear that SIMPOR performs better than others at the F1 score of 0.9170 (Figure 3b). Without any balancing techniques, it is not surprising that the classification result on the raw imbalanced data achieves the lower F1 Score, at 0.852 (Figure 3a). SMOTE technique does help to improve the classification F1-score to 0.862; however, it has not reached SIMPOR's performance. Other techniques perform poorly in this case; they could not achieve an F1-score as high as the raw data can achieve.

Additionally, SIMPOR, from our observation, results in a smooth and robust model decision boundary. We can see that the Random Over Sampling which randomly duplicates minority samples might cause overfitting where samples are duplicated many times and significantly increase their weights. SMOTE does better than Random Over Sampling. However, due to the fact that SMOTE does not take the informative region into account, unbalanced data in this area lead to a severe error in decision boundary. In Figures 3f and 3e, BorderlineSMOTE and ADASYN focus on the area near the model's decision boundary, but they inherit a drawback from SMOTE; any noise or mislabeled samples can create very dense bridges crossing the expected border so that it leads to decision errors. In contrast, by generating neighbors of samples in the direction towards the minority class and balancing the informative region, SIMPOR (Figure 3b) helps the classifier to make a better decision with a solid smooth decision line. It is also worth noticing that the classifier decision boundary lines of all other techniques are rougher than that of SIMPOR. This is because

Table 2: Classification results of different data balancing techniques on Breast Cancer Dataset.

Breast Cancer							
Majority:Minority (IR)	Metric	**SIMPOR**	**SMOTE**	**Borderline SMOTE**	**Over-Sampling**	**ADASYN**	**RawData**
357:120 (3:1)	F1	**0.953**	0.931	0.891	0.935	0.835	0.944
	AUC	**0.974**	**0.974**	0.964	**0.974**	**0.974**	0.971
	Precision	0.970	0.924	0.868	0.935	0.790	**0.975**
	Recall	**0.938**	**0.938**	0.922	0.936	0.928	0.919
357:50 (7:1)	F1	**0.943**	0.903	0.865	0.939	0.821	0.879
	AUC	**0.968**	**0.968**	0.967	0.967	**0.968**	0.966
	Precision	**0.952**	0.899	0.846	0.948	0.805	0.873
	Recall	**0.936**	0.918	0.903	0.932	0.879	0.907

they randomly generate synthetic samples, and it might cause data imbalance in the informative region.

6.3. SIMPOR on Breast Cancer dataset

6.3.1. Dataset

Breast Cancer is a real-world dataset containing information on cancer patients collected by Dua and Graff [7]. This has two classes including 357 negatives and 212 positives; each record includes 30 features extracted from a digitized image of a fine needle aspirate of a breast mass e.g., radius, area, and perimeter. Some of the records in the training dataset are randomly removed to create different imbalance ratios.

6.3.2. Results and Discussions

Table 2 shows the classification results of different data balancing techniques on the Breast Cancer dataset over the imbalance ratios of 3:1 and 7:1. Overall, SIMPOR outperforms other techniques on both imbalance ratios at an F1-score of 0.947 and 0.935 for IR of 3:1 and 7:1, respectively. SIMPOR also achieves the highest AUC results at 0.977 and 0.986 for both imbalance ratios. Both tests over the raw data (without any balancing technique) received the lowest F1-score as expected.

6.4. SIMPOR on Credit Card Fraud

6.4.1. Dataset

In this section, we experiment with our technique on Credit Card Fraud dataset [14]. The original dataset contains 492 fraud records out of 284,807 transactions; each record includes 30 features. During the experiments, we fix the size of fraud records as it becomes the minority class and randomly select a part of the remaining normal transactions to generate new datasets with different imbalance ratios. The classification model for the experiments under this dataset can be found in Table 1.

6.4.2. Results and Discussions

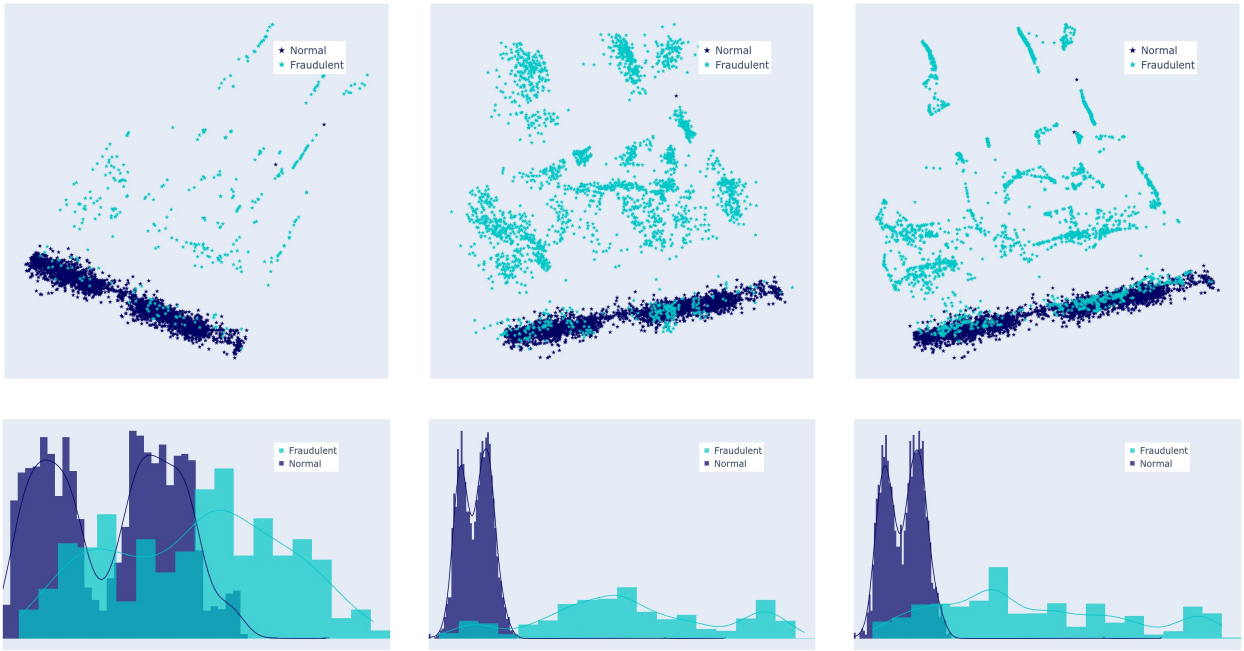
Table 3 shows the classification results on Credit Card Fraud dataset over different balancing techniques. SIMPOR constantly overperforms other techniques over all the settings and achieves the highest F1 and AUC score. While SIMPOR, SMOTE improve classification performance in most of the settings, ADASYN, BorderlineSMOTE fail to create good synthetic samples and reduce the classification performance.

To better understand how the techniques perform, we visualize the generated data by projecting them onto lower dimensions (i.e., one and two dimensions) space using the Principle Component Analysis technique (PCA) [10]. Data's 2-Dimension (2D) plots and 1-Dimension histograms are presented in Figure 4. A hard-to-differentiate ratio (HDR) is defined as the ratio of intersection between 2 classes in the 1D histogram to the total Fraudulent samples ($HDR = \frac{No.\ Intersection\ samples}{No.\ Fraud\ samples}$). This ratio is expected to be as small as 0% if the two classes are well separated; in contrast, 100% indicates that the two classes are unable to be distinguished. Other than HDR, the bottom tables in Figure 4 also show the absolute numbers of Fraudulent, Normal, and Intersection samples for each technique. From the plots, we can observe how the data distribute in 2D space and quantify hard-to-differentiate samples in the histograms.

While some techniques help to reduce the hard-to-differentiate ratios, others increase this ratio and worsen the data distribution. For example, in the 1D histogram of the ADASYN technique (Figure 4d), there are 2115 samples in the histogram intersection between 2 classes; they account for 81.82% of the total 2585 Fraudulent samples, which is worse than HDR of the Rawdata case (69.75%). In addition, the 2-Dimension plot illustrates that many synthetic samples (Fraudulent class) cross the majority (Normal class). Similarly, Figure 4e shows that BorderlineSMOTE also poorly generates synthetic minority data in this setting (CreditCard Fraudulent dataset with IR of 7:1) with an HDR of 75.98%. Among all techniques, SIMPOR achieves the best HDR of 11.14% and the synthetic data are far away from the majority class, which helps to improve the classification results as shown in Table 3.

7. Related Work

In the last few decades, there have been a number of solutions proposed to alleviate the negative impacts of data imbal-



Fraud	Normal	Inter.	HDR
367	2585	256	69.75%

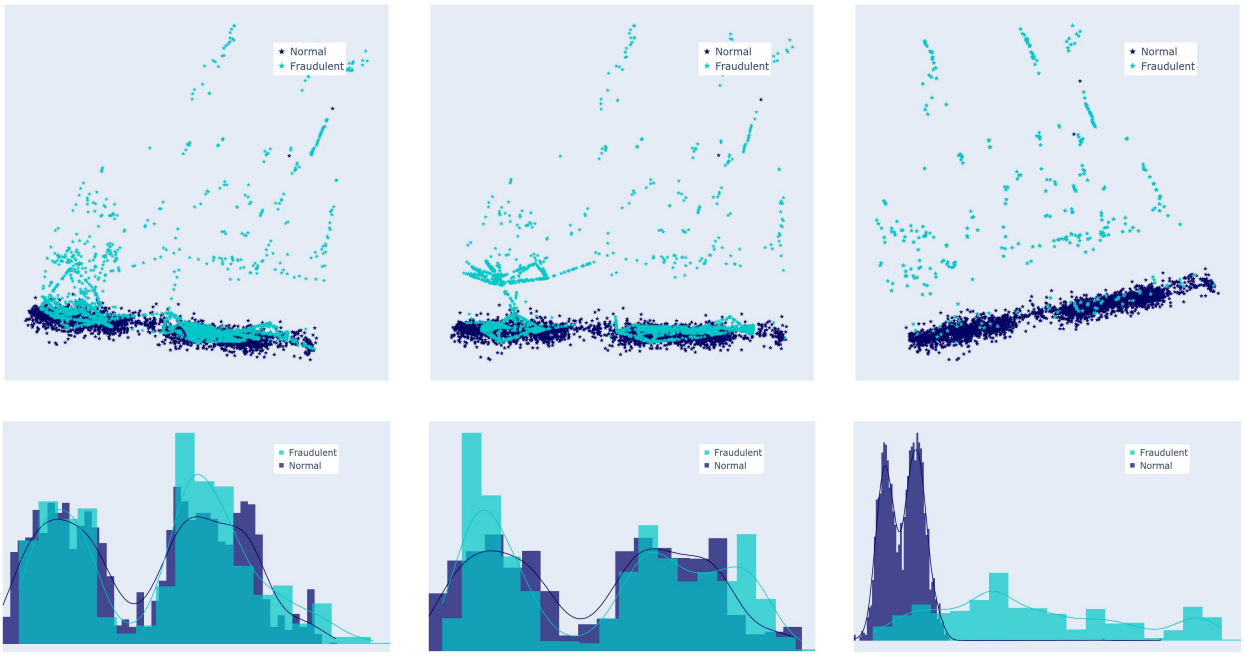
(a) Raw Data.

Fraud	Normal	Inter.	HDR
2585	2585	288	**11.14**%

(b) Generated data by SIMPOR.

Fraud	Normal	Inter.	HDR
2585	2585	534	20.66%

(c) Generated data by SMOTE.



Fraud	Normal	Inter.	HDR
2585	2585	2115	81.82%

(d) Generated data by ADASYN.

Fraud	Normal	Inter.	HDR
2585	2585	1964	75.98%

(e) Generated data by BorderlineSMOTE.

Fraud	Normal	Inter.	HDR
2585	2585	544	21.4%

(f) Generated data by Oversampling.

Figure 4: Generated training data projected onto 2-dimension space and their histograms in 1-Dimension space using Principle Component Analysis dimension reduction technique. The explanation tables illustrate the number of samples in each class (Fraudulent and Normal), 1-Dimension histogram intersection between 2 classes, and the hard-to-differentiate ratio ($HDR = \frac{Inter.}{Fraud} 100\%$).

ance in machine learning. However, many of them are not efficient when it comes to high-dimensional data and deep learning. In this section, we review algorithms that aim at deep learning and strategies inherited from conventional machine learn-

Table 3: Classification results of different data balancing techniques on Credit Card Fraud Dataset.

Credit Card Fraud							
Majority:Minority (IR)	Metric	**SIMPOR**	**SMOTE**	**Borderline SMOTE**	**Over-Sampling**	**ADASYN**	**RawData**
1470:490 (3:1)	F1	**0.943**	0.903	0.865	0.939	0.821	0.879
	AUC	**0.968**	**0.968**	0.967	0.967	**0.968**	0.966
	Precision	**0.952**	0.899	0.846	0.948	0.805	0.873
	Recall	**0.936**	0.918	0.903	0.932	0.879	0.907
3430:490 (7:1)	F1	**0.953**	0.931	0.891	0.935	0.835	0.944
	AUC	**0.974**	**0.974**	0.964	**0.974**	**0.974**	0.971
	Precision	**0.975**	0.924	0.868	0.935	0.790	**0.975**
	Recall	**0.938**	**0.938**	0.922	0.936	0.928	0.919
4900:490 (10:1)	F1	**0.952**	0.901	0.863	0.658	0.906	0.930
	AUC	**0.972**	0.957	0.967	0.964	0.969	0.975
	Precision	**0.979**	0.891	0.855	0.662	0.885	0.931
	Recall	0.929	0.925	0.918	0.845	0.933	**0.936**
7350:490 (15:1)	F1	**0.952**	0.909	0.885	0.917	0.883	0.944
	AUC	**0.968**	0.963	0.955	0.963	0.965	0.960
	Precision	0.966	0.892	0.850	0.909	0.849	**0.984**
	Recall	**0.940**	0.931	0.931	0.935	0.931	0.911

ing methods. These techniques can mainly be categorized into two different categories, i.e., data-centric and model-centric approaches.

Model-centric approaches usually require modifications of algorithms on the cost functions in order to balance the weight of each class. Specifically, such cost-sensitive approaches put higher penalties on majority classes and less on minority classes to balance their contribution to the final cost. For example, [5] provided their designed formula $(1-\beta^n)/(1-\beta)$ to compute the weight of each class based on the effective number of samples n and a hyperparameter β which is then applied to re-balance the loss of a convolutional neural network model. [18],[27], [23] assign classes' weights inversely proportional to sample frequency appearing in each class.

Compared to model-centric-based manners, data-centric approaches have been attracting more research attention as it is independent of machine learning algorithms. In this category, we divide into two main approaches, i.e. sampling-based and generative approaches. Sampling-based methods [31], [11], [15], [20], [9] mainly generate a balanced dataset by either oversampling minority classes or down-sampling majority classes. Some methods are not designed for deep learning, but we still consider them since they are independent of the machine learning model architecture. In a widely used method SMOTE [4], Chawla *et al.* attempt to oversampling minority class samples by connecting an sample to its neighbors in feature space and arbitrarily drawing synthetic samples along the connections. However, one of the drawbacks of SMOTE is that if there are samples in the minority class located in the majority class, it creates synthetic sample bridges towards the majority class [13]. This renders difficulties in differentiation between the two classes. Another SMOTE-based work namely Borderline-SMOTE [16] was proposed in which its method aims to do SMOTE with only samples near the border between classes. The samples near the border are determined by the labels of its k distance-based neighbors (if more than half of neighbors belong to the other class, the sample is considered to be on the border). This "border" idea is similar to ours to some degree. However, finding a good k is critical, and it is usually highly data-dependent. In addition, Borderline-SMOTE again faces the problems of SMOTE.

Under the down-sampling category, other works [8], [3] leverage active learning techniques to find informative samples which authors believe the imbalance ratio in these areas is much smaller than that in the entire dataset. They then classify this small pool of samples to improve the performance and expedite the training process for the SVM-based method. however, this method was only designed for SVM-based methods which mainly depend on the support vectors. Also, this potentially discards important information of the entire dataset because only a small pool of data is used.

Generative approaches which generate synthetic samples in minor classes by sampling from data distribution are becoming more attractive as they are outperforming other methods in high dimensional data [21]. When it comes to images, a number of deep learning generative-based methods have been proposed as deep learning is capable of capturing good image representations. [28] [6] [24] utilized Variational Autoencoder as a generative model to arbitrarily generate images from learned distributions. However, most of them assumed simple prior distributions such as Gaussian for minor classes, they tend to simplify data distribution and might not succeed in sophisticated distri-

butions. Our solution also falls into this category; however, we leverage the idea of a mixture model to tackle this issue for image data.

8. Conclusion

We propose a data balancing technique by generating synthetic data for minority samples, which maximizes the posterior ratio to embrace the chance they fall into the minority class and do not fall across the expected decision boundary. While maximizing the posterior ratio, we use kernel density estimation to estimate the likelihood so that it is able to work with complex distribution data without requiring data distribution assumptions. In addition, our technique leverage entropy-based active learning to find and balance the most informative samples. This is important to improve model performance as we have shown in our experiments. In future work, we would like to investigate imbalanced image datasets and enhance our technique to adapt to image data.

Acknowledgment

Efforts sponsored in whole or in part by United States Special Operations Command (USSOCOM), under Partnership Intermediary Agreement No. H92222-15-3-0001-01. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. [1]

References

[1] Maximal and minimal points of functions theory.

[2] Sphere, Aug 2021.

[3] Umang Aggarwal, Adrian Popescu, and Celine Hudelot. Active Learning for Imbalanced Datasets. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1417–1426, Snowmass Village, CO, USA, March 2020. IEEE.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002.

[5] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-Balanced Loss Based on Effective Number of Samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9260–9269, Long Beach, CA, USA, June 2019. IEEE.

[6] Wangzhi Dai, Kenney Ng, Kristen Severson, Wei Huang, Fred Anderson, and Collin Stultz. Generative Oversampling with a Contrastive Variational Autoencoder. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 101–109, Beijing, China, November 2019. IEEE.

[7] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[8] Seyda Ertekin, Jian Huang, Leon Bottou, and Lee Giles. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management - CIKM '07*, page 127, Lisbon, Portugal, 2007. ACM Press.

[9] Seyda Ertekin, Jian Huang, and C. Lee Giles. Active learning for class imbalance problem. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '07*, page 823, Amsterdam, The Netherlands, 2007. ACM Press.

[10] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[11] Yonatan Geifman and Ran El-Yaniv. Deep active learning over the long tail. *CoRR*, abs/1711.00941, 2017.

[12] Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning, 2019.

[13] Dr Saptarsi Goswami. Class Imbalance, SMOTE, Borderline SMOTE, ADASYN, November 2020.

[14] Gokhan Goy, Cengiz Gezer, and Vehbi Cagri Gungor. Credit Card Fraud Detection with Machine Learning Methods. In *2019 4th International Conference on Computer Science and Engineering (UBMK)*, pages 350–354, Samsun, Turkey, September 2019. IEEE.

[15] Haibo He and E.A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, September 2009.

[16] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang, editors, *Advances in Intelligent Computing*, pages 878–887, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

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

[18] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning Deep Representation for Imbalanced Classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5375–5384, Las Vegas, NV, USA, June 2016. IEEE.

[19] S. Leroueil. Compressibility of Clays: Fundamental and Practical Aspects. *Journal of Geotechnical Engineering*, 122(7):534–543, July 1996.

[20] Lusi Li, Haibo He, and Jie Li. Entropy-based Sampling Approaches for Multi-Class Imbalanced Problems. *IEEE Transactions on Knowledge and Data Engineering*, 32(11):2159–2170, November 2020.

[21] Alexander Liu, Joydeep Ghosh, and Cheryl E. Martin. Generative oversampling for mining imbalanced datasets. In Robert Stahlbock, Sven F. Crone, and Stefan Lessmann, editors, *Proceedings of the 2007 International Conference on Data Mining, DMIN 2007, June 25-28, 2007, Las Vegas, Nevada, USA*, pages 66–72. CSREA Press, 2007.

[22] Yang Liu, Xiang Li, Xianbang Chen, Xi Wang, and Huaqiang Li. High-Performance Machine Learning for Large-Scale Data Classification considering Class Imbalance. *Scientific Programming*, 2020:1–16, May 2020.

[23] Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. *CoRR*, abs/1805.00932, 2018.

[24] Sankha Subhra Mullick, Shounak Datta, and Swagatam Das. Generative Adversarial Minority Oversampling. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1695–1704, Seoul, Korea (South), October 2019. IEEE.

[25] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. Cuda, release: 10.2.89, 2020.

[26] Zhicong Qiu, David J. Miller, and George Kesidis. A maximum entropy framework for semisupervised and active learning with unknown and label-scarce classes. *IEEE Transactions on Neural Networks and Learning Systems*, 28(4):917–933, 2017.

[27] Vivek Kumar Rangarajan Sridhar. Unsupervised Text Normalization Using Distributed Representations of Words and Phrases. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 8–16, Denver, Colorado, 2015. Association for Computational Linguistics.

[28] M.H. Rashid, J.H. Wang, and C. Li. Convergence analysis of a method for variational inclusions. *Applicable Analysis*, 91(10):1943–1956, October 2012.

[29] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08*, page 1070, Honolulu, Hawaii, 2008. Association for Computational Linguistics.

[30] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, July 1948.

[1] The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the United States Special Operations Command.

[31] Li Shen, Zhouchen Lin, and Qingming Huang. Learning deep convolutional neural networks for places2 scene recognition. *CoRR*, abs/1512.05830, 2015.

[32] Qian Ya-Guan, Ma Jun, Zhang Xi-Min, Pan Jun, Zhou Wu-Jie, Wu Shu-Hui, Yun Ben-Sheng, and Lei Jing-Sheng. EMSGD: An Improved Learning Algorithm of Neural Networks With Imbalanced Data. *IEEE Access*, 8:64086–64098, 2020.