2008 Eighth IEEE International Conference on Data Mining

Start Globally, Optimize Locally, Predict Globally: Improving Performance on Imbalanced Data

David A. Cieslak, Nitesh V. Chawla
Department of Computer Science and Engineering, University of Notre Dame, USA
{dcieslak,nchawla}@cse.nd.edu

Abstract

Class imbalance is a ubiquitous problem in supervised learning and has gained wide-scale attention in the literature. Perhaps the most prevalent solution is to apply sampling to training data in order improve classifier performance. The typical approach will apply uniform levels of sampling globally. However, we believe that data is typically multi-modal, which suggests sampling should be treated locally rather than globally. It is the purpose of this paper to propose a framework which first identifies meaningful regions of data and then proceeds to find optimal sampling levels within each. This paper demonstrates that a global classifier trained on data locally sampled produces superior rank-orderings on a wide range of real-world and artificial datasets as compared to contemporary global sampling methods.

1 Introduction

Mining imbalanced datasets continues to be a pervasive problem in a large variety of applications [8, 22] including medicine, finance, and security. Traditional objective metrics, such as accuracy, often fail to describe performance properly under these circumstances as a trivial classifier may produce high accuracy but perform poorly on the more interesting rare class. Instead, Area Under the ROC Curve ($AUROC$) is typically used as a measure of the capability of a classifier to effectively rank the minority or positive class instances above the majority or negative class instances. $AUROC$ captures the trade-off between true positives and false positives, producing a robust metric for even the most imbalanced datasets.

The most significant direction of research on imbalanced datasets has focused on sampling strategies for countering the problem of class imbalance [3, 8]. Sampling methods focus on either adding minority (positive) class points or removing majority (negative) class points to improve accuracy or $AUROC$ of the positive class [3, 15]. Sampling is a common approach to counter the curse of imbalance in many domains.

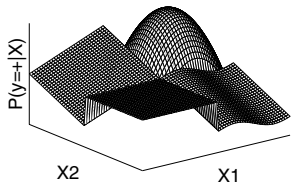

Figure 1. Data example for which the class distributions are generated through piecewise, regional functions.

Sampling methods consider the class skew and properties of the dataset as a whole. However, it is the contention of this paper that machine learning and data mining often face nontrivial datasets, which often exhibit characteristics and properties at a local, rather than global level. As an example, we consider Figure 1 where the class probability generating function is different within each region of the data, dictating that the relationship between classes is substantially different within each discrete area. Even when two regions exhibit the same class skew, it is possible for them to exhibit other significant and disjoint characteristics. While a particular sampling level induces a strong classifier for the first region, it may well induce a poor classifier on the second region. This suggests that the typically global task of selecting sampling levels to improve classifier performance should in many contexts be considered locally.

With these concerns in mind, the primary **contributions** of this paper are as follows. 1) An analytic framework for supporting locally optimal mixtures of sampling . 2) An effective data partitioning or segmentation method based on the distribution invariant Hellinger distance [9, 10, 11, 21]. 3) Identifying optimal levels and types of sampling within each data segment. 4) A comprehensive evaluation using 20 real-world and UCI datasets.

1550-4786/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDM.2008.87

For benchmarking, we will consider two popular sampling methods from the related work: Synthetic Minority Oversampling TechniquE (SMOTE) [6] and random undersampling. In addition, we include Classification using lOcal clusterinG Over Sampling (COG-OS) [23]. SMOTE injects synthetic positive class examples to emphasize class boundaries, while COG-OS uses k-means clustering to partition the complex majority class into simpler sub-classes and minority class replication.

2 Start Globally, Optimize Locally

In the supervised learning task a labeled training data sample D is available for model training, presumably drawn randomly from some distribution $p(Y|X)$, for which each $x_i \in X$ is a single feature vector and $y_i \in Y$ is the associated class label. In addition, there is an available unlabeled sample T for testing purposes, which for the sake of this study is presumed from the same distribution $p(Y|X)$. The goal of this task is then to train some classifier $f : X \longmapsto Y$ to estimate the probability for each x_i in T to belong to a class in Y. The ideal for this classifier is strong performance, either in minimizing loss or cost or maximizing accuracy, profit, or $AUROC$. Thus, the model should maximize some objective function, O, such that $E_{(X,Y)}[O(f(X), Y)]$ is maximized [4]. In the ideal case, $f : X \longmapsto Y$ represents the "true model" of the data $p(Y|X)$. In practice, this is difficult if not impossible due to the limited representation D provides of $p(Y|X)$. Thus, the learning algorithms often approximate the true function. Due to the bias inherent in each learning algorithm's representation and the particular emphasis of each objective function, it is often the case that D^*, a resampling of D not necessarily drawn from $p(Y|X)$, will actually enhance the performance of the determined classifier. The task of finding some $S : D \longmapsto D^*$ that improves a classifier's performance is called sampling.

A primary question when employing sampling is *how does one tune $S : D \longmapsto D^*$ to yield optimal results?* A suggested method is to use a heuristic wrapper [7] to set sampling levels for each class. The wrapper greedily explores the search space of sampling parameters until the "potentially optimal" amounts are discovered. An alternate method stems from cost sensitive learning. For instance, Elkan discusses a simple method to calculate optimal sampling levels based on the ratio of misclassification for each class [14]. However, evaluation typically occurs without explicit costs. In this case, Elkan's calculation simply indicates to sample the classes to a balance point, which can be limiting. It is often necessary to explore a much larger sampling space, to address the question of *how do the properties of data (aside from class skew) affect optimal sampling levels?* The wrapper exploits the properties of the data as it utilizes the performance of the learning algorithm as a guide.

We consider this question by constructing some artificial datasets, seen in Figure 2, as follows. To begin, there exist two classes of examples with a skew ratio $(+ : -)$ of $(50 : 1000)$, positive examples depicted as $+$'s and negative examples as $\cdot$'s. Each class is centered at a fixed point $\bar{x}_+$ and $\bar{x}_-$ with radius r_+ and r_-, respectively. Since the number of points for each class is fixed, the densities ρ_+ and ρ_- are a product of the radii, reducing radius length increases density. The final consideration for this dataset is δ, the distance between $\bar{x}_+$ and $\bar{x}_-$. The higher the δ, the lower the class overlap. Of interest is the interaction between the density ρ_+, relative overlap between the regions, and end ability to discriminate between $+$ and $-$ as quantified by $AUROC$. Since maintaining the relative skew is desired, ρ_+ will be affected by changing r_+ and the resultant region.

This experiment considers three discrete increments of both r_+ and δ: High, Medium, and Low to determine the importance of degree of r_+ and δ on end performance. We say that ρ_+ is High when $r_+ = r_-$, Medium when $r_+ = 0.75r_-$, and Low when $r_+ = 0.5r_-$. Once r_+ is determined, overlap is High when $\delta = 0.5r_+$, Medium when $\delta = 0.75r_+$ and Low when $\delta = r_+$. When overlap is High, nearly all of the $+$ region is engulfed within the $-$ region. However, when overlap is Low, over half the $+$ region is outside the larger $-$ area. We note that $\bar{x}_-$ and ρ_- (and therefore r_-) remain fixed through all constructions. Based on these constructions, we now address *how, if at all, do these constructed properties affect discriminative ability?*

As we have conserved the relative skew ratio between classes at $(50 : 1000)$, Elkan's solution would apply the same level of over and undersampling as a universal optimal solution to each of the nine scenarios. However, the results shown in Table 1 indicate otherwise, also establishing three general trends. First, increased minority class density improves $AUROC$, since increased density allows C4.5 to develop a more confident decision region. Second, increased separation between the two class centers (reduced class overlap) increases $AUROC$, since this enables C4.5 to construct a more definitive border between the classes. Third, the optimal levels of sampling for each scenario is substantially different (and does not generate a class balanced sample), which is in contradiction with Elkan's solution. Thus, it is observed that data properties aside from class skew may have a substantial effect on optimal sampling level.

However, datasets are often more complex than those in Figure 2, perhaps containing multiple points of intersection. In such a scenario, even an optimized sampling of data may represent an incomplete solution from a local perspective. Thus, we would like to address *what effect*

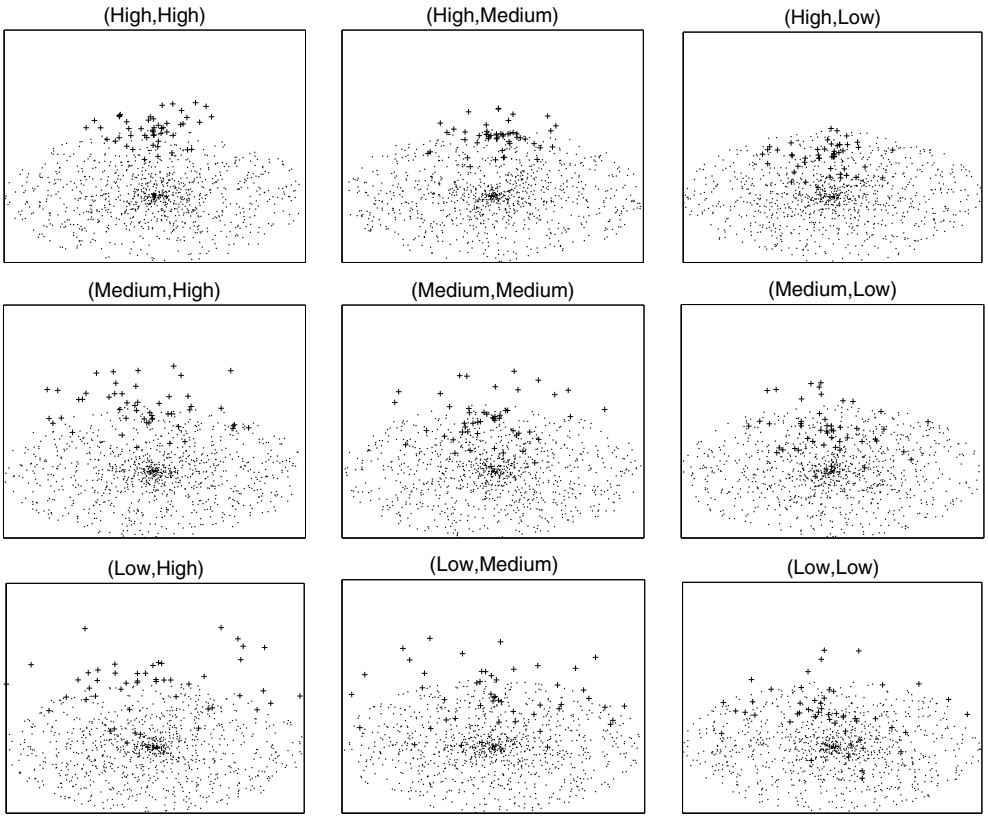

Figure 2. Demonstration of the different two-class radial distributions, with pairs (ρ, δ) specifying the level of density ρ and centroid separation δ for each. As the rows descend, ρ decreases and δ likewise decreases from the left to right columns. For example, (High,Medium) displays the dataset with High density and Medium class separation. The $(+:-)$ skew in this situation is $(50:1000)$. Each $+$ represents a positive example, while each $\cdot$ represents a negative example.

does considering sampling locally have on global performance? We consider the data examples highlighted in Figure 3. We note that the left and middle datasets are similar to the previous: they represent two-class overlapping radials, with varying distance between centers and density. The left dataset is significantly easier to learn than the middle, since there is less overlap between the regions. This is reflected by the determined $AUROC$ for left and middle of 0.868 and 0.531, using C4.5. Using sampling on the left dataset, the $AUROC$ can be improved to 0.906 using levels of $(Undersample, SMOTE)$ of $(50, 350)$ found through the wrapper method in [7], while the middle can improve to 0.812 using $(20, 250)$. However, there is some difficulty when the left and middle dataset concatenate to form the rightmost dataset in the figure. By default, C4.5 obtains an $AUROC$ of 0.817, which can be improved to 0.845 using $(100, 50)$. This is somewhat problematic. The components dictated different sampling levels, but when they were considered as a single dataset a distinct third sampling combination was used, indicating that there is a loss in efficiency of optimizing $AUROC$ via sampling globally. This is confirmed, since using the regional sampling level of $(50, 350)$ for the leftmost dataset and $(20, 250)$ for the center dataset improves global performance further to 0.874. Thus, our current methods of global sampling may be insufficient in some applications, since we observe a considerable improvement when local sampling is considered.

Rather, the task of finding some $S : D \longmapsto D^*$ to improve performance might be divided into components. It is our belief that much of data operates as a set of modalities guiding the construction of feature vectors and their associated class labels, in accordance with $P(Y|X)$.

Presuming at least some data exists in such a modal fashion, it is therefore meaningful to derive samplings locally. Such a procedure begins by decomposing D into m meaningful segments, D_i. In Figure 3, the two components (D_1, D_2) for the right dataset are the left and middle datasets. Once such segments have been determined, sam-

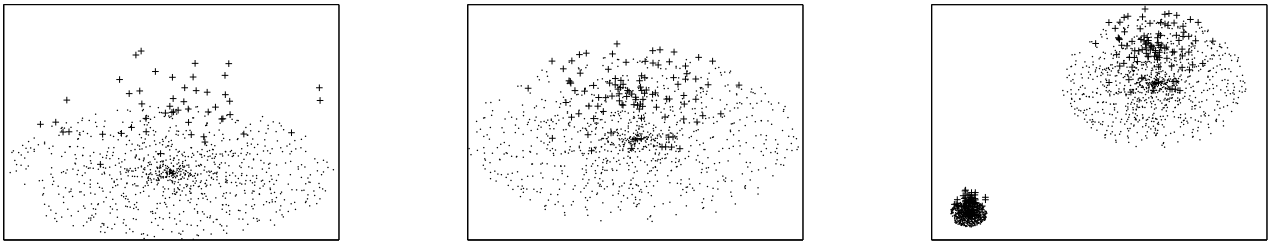

Figure 3. Two separate two class radials with different δ,ρ_+, and skews. The third reflects their combination into a single dataset.

Table 1. $AUROC$ values for C4.5 and C4.5 with Sampling, in addition to optimal sampling levels, on each of the nine scenarios in Figure 2.

(ρ_+, δ)	$AUROC$ $(C45, C45+S)$	$(Undersampling, SMOTE)$
(High,High)	(0.926, 0.968)	(60, 450)
(High,Medium)	(0.909, 0.942)	(40, 250)
(High,Low)	(0.898, 0.904)	(40, 100)
(Medium,High)	(0.915, 0.961)	(50, 300)
(Medium,Medium)	(0.878, 0.927)	(60, 250)
(Medium,Low)	(0.814, 0.831)	(30, 250)
(Low,High)	(0.892, 0.940)	(100, 150)
(Low,Medium)	(0.825, 0.847)	(30, 350)
(Low,Low)	(0.705, 0.736)	(80, 500)

pling optimizes locally to find each D_i^*. This is recombined as $D^* = \bigcup_{i=1}^{m} D_i^*$, which is then used to formulate a global classifier. As can be seen from this example, a key aspect to local optimization is constructing a reliable partitioning which will be discussed in the next section.

3 Partitioning Data

This section introduces a supervised method for splitting the data into segments or components. The partitioning builds a tree of maximum height 2 (see Algorithm 1), resulting in at most 4 segments. Our initial experimentation demonstrated that using a height greater than 2 resulted in much sparser segments, given the high imbalance of some of the datasets used in our paper. Thus, to be consistent we constrained the tree to the height of 2 for all datasets. This type of segmentation is used instead of unsupervised learning, such as k-means, since the known class labels provide potentially useful information. The data will also be presumably be highly imbalanced, and we want to exploit the class skew information while constructing the partitions as well. To that end, we propose using Hellinger distance [9, 10, 11, 21] to guide the segmentation.

Hellinger distance is a measure of distributional divergence. Let (Θ, λ) denote a measurable space with A and B as two continuous distributions measured with respect to λ. Let a and b be the densities of A and B with respect to λ. The definition of Hellinger distance can be given as:

$$d_H(A, B) = \sqrt{\int_\Omega (\sqrt{a} - \sqrt{b})^2 d\lambda} \tag{1}$$

This is equivalent to:

$$d_H(A, B) = \sqrt{2(1 - \int_\Omega \sqrt{ab} d\lambda)} \tag{2}$$

where $\int_\Omega \sqrt{ab} d\lambda$ is the Hellinger integral. Note the Hellinger distance does not depend on the choice of parameter λ. It can also be defined for a countable space Φ, as $d_H(A, B) = \sqrt{\Sigma_{\phi \in \Phi}(\sqrt{A(\phi)} - \sqrt{B(\phi)})^2}$. The Hellinger distance carries these following properties: 1) $d_H(A, B)$ is in $[0, \sqrt{2}]$. 2) d_H is symmetric and non-negative, implying $d_H(A, B) = d_H(B, A)$. Moreover, squared Hellinger distance is the lower bound of KL divergence.

Here, the A and B in Equation 1 are assumed to be the normalized frequencies of feature values across classes. This allows us to capture the notion of "affinity" between the probability measures on a finite event space. If A = B, then distance = 0 (maximal affinity) and if A and B are completely disjoint then distance = $\sqrt{(2)}$ (zero affinity). This dictates the partition splitting criterion for separability between classes. We want to select a feature that carries the minimal affinity between the classes.

We assume a countable space, so we discretize all continuous features into p partitions or bins. Then, we are essentially interested in calculating the "distance" between the relative class frequencies for each feature value, normalized by the overall class frequency to reduce the effect of class skew. Thus, we apply Hellinger distance as

Algorithm 1 Hellinger Partitioning Procedure

Input: Dataset D with feature set F
Current height h of the binary tree (called with 0)
1: **if** $h == 2$ or $|D| == 0$ **then**
2: **return**
3: **end if**
4: $f = \text{argmax}_{i \in F}\, d_H(T_+, T_-)$
5: **for** each value **v** of f **do**
6: $HellingerPartition(T_{f=v}, h+1)$
7: **end for**

$$d_H(+,-) = \sqrt{\sum_{j=1}^{p} \left(\sqrt{\frac{P(+|X=x_j)}{P(+)}} - \sqrt{\frac{P(-|X=x_j)}{P(-)}} \right)^2} \quad (3)$$

We note that bounds on this metric are unaffected by the class skew, making Hellinger distance ideal for separating regions of disjoint class distribution within highly imbalanced data. To construct partitions, we use the procedure given in Algorithm 1. The Hellinger distance is applied to each feature with the highest distance used to generate a split in data to generate a two-level binary tree, as initial experimentation indicated this produced the best results. In the case of continuous features each potential feature split is explored, meaning $p = 2$. The procedure is then applied on each partition until a desired maximum depth is reached. With this partitioning method, we now propose a generalized framework for local sampling aimed to leverage the observations of Section 2 into practical application.

4 Applying Local Sampling

Local Sampling (LS) provides a general framework into which it is possible to incorporate any sort of classifier, data partitioning, or sampling mechanism. This flexible structure allows for versatility, allowing a practitioner to readily adapt this framework to many application settings. In essence, this is a three step extension to any sampling scheme.

Algorithm 2 outlines the LS pseudo-code. While we use the proposed Hellinger distance based partitioning method, in principle any arbitrary partitioning method P is applied to divide the data. This phase may apply nearly any type of supervised or unsupervised method. With the set of discovered data partitions Z, the procedure moves to the sampling phase. Here, the chosen sampling method S is applied to each component. The algorithm checks the range of parameters θ for S and determines the optimal performer according to the objective function O. Once optimal sampling parameters are applied to each component or segment, all the components are merged to create the new training set. Then,

Algorithm 2 The Local Sampling (LS) Framework

Input: D: Labeled Training Set
S Sampling Method with associated parameters θ
P Partitioning Method finding m partitions
f Learning Algorithm
O Objective Function
Output: M: the model built
Z is the data segments or partitions discovered via P.
1: $Z = P(D)$
2: **for** each partition $z \in Z$ **do**
3: $D_z^* = \text{argmax}_{\theta_i \in \theta}\, O(S(z, \theta_i))$
4: **end for**
5: $D^* = \bigcup_{i=1}^{m} D_i^*$
6: $M = f(D^*)$

a classifier is learned. Thus, we go from global view of the data to local view (partitioning) and then again to a global view for learning the classifier.

We follow the wrapper framework [7] to discover the potentially optimal amounts of undersampling and SMOTE for S. Note that the same wrapper framework is used for the benchmarks used in this paper – COG-OS and combination of undersampling and SMOTE on the entire dataset (without LS). The sampling levels for each form the parameters θ and are $\{0, 10, ...100\}$ for undersampling and $\{0, 50, 100, ..., 500\}$ for SMOTE. For P, we apply the partitioning method from Section 3. The wrapper then proceeds as follows. First, a 5-fold cross-validation (cv) is applied on the training set to determine the baseline $AUROC$ performance. Since, we use 5x2 cross-validation framework, this 5-fold cv is applied to each of the training sets of 5x2, making it an independent validation framework. Then, the majority class examples are randomly removed 10% at a time, as long as the performance remains above the baseline. Once an undersampling level is selected, it results in the new improved baseline performance. Subsequently, SMOTE is applied in progressive steps, adding synthetic points in increments of 50% of the minority class size. Again, this is repeated as long as it beats the baseline performance. This procedure is slightly adapted for local sampling by exploring the parameter space exhaustively for each partition in sequence. The order of optimization is the partition with the most majority class points to the one with the fewest, allowing us to gradually shift the bias towards the minority class. It is also likely that not all partitions will require either or both undersampling and SMOTE. The partitions are merged after each optimization to determine the $AUROC$ via the 5-fold cv, allowing for measurement of the global relevance to any optimization done on the partition.

Figure 4 depicts the practical application and benefits of LS on the Pima dataset [2], represented in two dimensions. While the $+$ and $-$ classes have disjoint regions, a signifi-

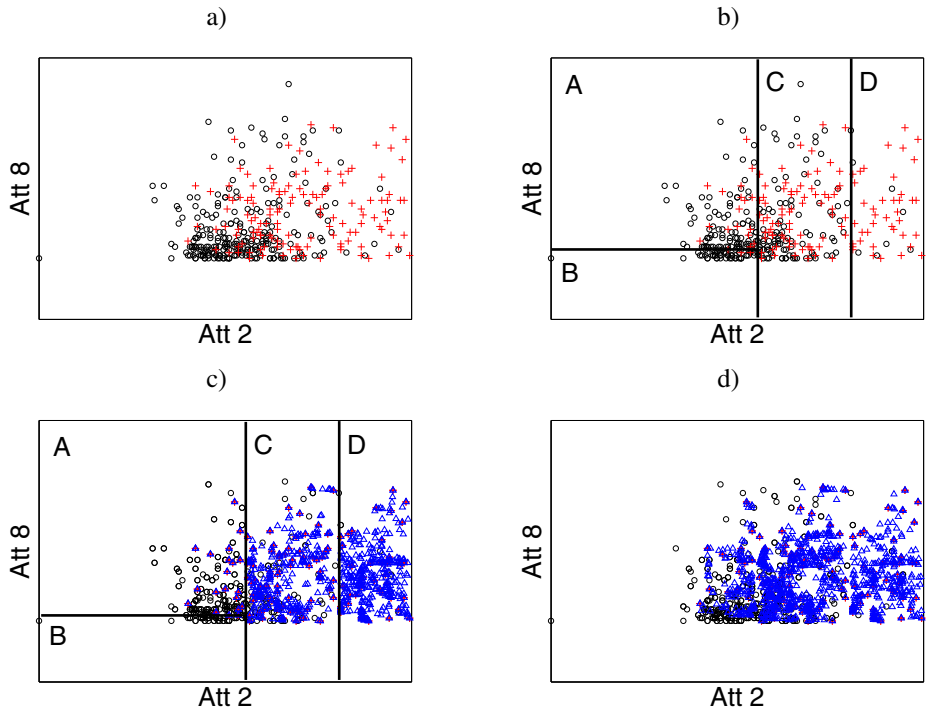

Figure 4. (a) The Pima dataset, illustrated on two dimensions. $+$ represents positive examples, while $\bigcirc$ represents negative examples. (b) Using Hellinger distance, a two-level tree is trained and then combinations of sampling levels are explored. (c) Each component optimizes to a different sampling level, removing some negative examples and injecting SMOTE examples represented by $\triangle$. The concatenation of components is used to form a single training set. This is juxtaposed with (d), which presents the effects of global undersampling and SMOTE.

cant class overlap is also observable in other regions. Using C4.5 alone yields an $AUROC$ of 0.765.

In Figure 4(b), Hellinger distance (discussed in Section 3) is used to form a four-region split of the dataset. Region B is entirely comprised of $-$ examples and A exhibits $-$ example dominance. However, regions C and D have isolated regions in which $+$ examples actually form the majority of examples. Using $AUROC$, each region is optimized with $(Undersample, SMOTE)$ levels of $(40, 50)$, $(90, 100)$, $(40, 250)$, and $(50, 500)$ for A, B, C, and D, respectively, seen in Figure 4(c). Doing so produces an $AUROC$ of 0.817, which outperforms a global undersampling and SMOTE combination, which finds an $AUROC$ of 0.790. The global sampling uses $(70, 400)$, while the local sampling effectively applies $(66, 303)$ (based on the number of examples actually added and removed for each class across all partitions). We have therefore observed a practical case similar to Figure 3, where LS improves $AUROC$. In this case, LS is more judicious in terms of requiring fewer examples for better results as it is able to focus use of undersampling and SMOTE to particular regions of the data.

5 Empirical Evaluation

In this section, we present experimental results to validate and understand the performance of LS on imbalanced datasets, particularly in comparison with other contemporary methods.

5.1 Performance Analysis

We use 5x2-fold cross-validation (cv) [13] with $AUROC$ as the evaluation criterion. We chose 5x2 cv over 10-fold cv as the use of 10-fold cv, while resulting in independent testing sets, carries sufficient overlap in training data across folds. This makes application of typical t-tests assuming independent trials questionable. Dietterich [13] notes that this results in an elevated level of Type I error, which can be corrected for by his 5x2 cv. This relies on the idea that learning curves rarely cross for algorithms as training set size varies. The elevated Type I error be-

comes of particular concern when evaluating on imbalanced datasets because of the trade-off between false positive and false negative. Significance will be tested by the F-test as described by Alpaydin [1] at 95% confidence. We note that two datasets, Compustat and Intrusion, possess natural train/test split which are maintained to preserve the sanity of evaluation. Significant differences in results will not be determined on these datasets.

We also use the statistical analysis framework to compare classifier ranks outlined by Demsar [12]. This method uses the Holm procedure of the Friedman test to establish the significance of classifier rankings across multiple data sets. This is a specialized, non-parametric procedure for testing the significance of differences between multiple means. Demsar notes that the Friedman test is "appropriate since it assumes some, but limited commensurability [12] (page 27)." It is "safer than parametric tests since it does not assume normal distributions or homogeneity of variance. As such, it can be applied to classification accuracies, error ratios or any other measure for evaluation of classifiers [12] (page 27)." Both 5x2 cv and Demsar's analysis are increasingly embraced as standards for classifier comparison in the data-mining community.

Table 2. All the datasets used in this paper. The Examples column for Compustat and Intrusion also includes the number of both training and testing examples.

No.	Dataset	Examples	Features	MinClass
1	Boundary	3,505	174	4%
2	Breast-W	569	30	33%
3	Calmodoulin	18,916	131	5%
4	Compustat	10,358/3,258	20	4%/4%
5	E-State	5,322	12	12%
6	FourClass	862	2	36%
7	German.Numer	1,000	24	30%
8	Intrusion	17,600/17,400	41	0.7%/0.8%
9	Letter	20,000	16	19%
10	Mammography	11,183	6	2%
11	Oil	937	49	4%
12	Page	5,473	10	10%
13	Pendigits	10,992	16	10%
14	Phoneme	5,404	5	21%
15	PhosS	11,411	479	5%
16	Pima	768	8	35%
17	Satimage	6,435	36	10%
18	Segment	2,310	19	16%
19	Splice	1,000	60	4.8%
20	SVMGuide1	3,089	4	35%

5.2 Methods and Benchmarks

We compare the proposed local sampling framework to other contemporary methods, notably the combination of undersampling and SMOTE, which has been shown to significantly improve performance on imbalanced datasets, and COG-OS [23], which is a recent method for countering class imbalance using clustering analysis. COG-OS uses local clusterings of majority class points and minority class replication to enhance linear separability. These clusterings are derived using k-means. Since this algorithm is partially dependent on the initial random seedings of centroids, we used 10 different clusterings, and the clustering resulting in the lowest distance to centroid was preserved.

Each of the sampling methods and COG-OS utilize the same framework for discovering the potentially optimal parameters allowing for fair comparisons across the board. That is, for the vanilla SMOTE and undersampling combination with C4.5, we used a 5-fold cv on the training fold of 5x2 cv to first determine the levels of sampling. For COG-OS, we empirically determined the appropriate number of clusters k, using the same 5-fold cv on top of 5x2 cv framework. Finally, for our LS approach we also used the 5-fold cv to determine the levels of sampling in each partition. The 5x2 training and testing sets remained the same across all the approaches. This ensured that all approaches were independently optimized, in the same fashion, to achieve their best performances.

We use two base classifiers: C4.5 [19] and SVM [5]. Here, decision trees are unpruned and use Laplace-smoothing at the leaves in keeping with the observations of Provost & Domingos [18] that this provides improvements for $AUROC$. We train the SVM using the parameters $-t\ 0\ -b\ 1$ and build a probabilistic estimator. Since, SVM training has a much higher time complexity than decision tree induction, which is exacerbated to a prohibitive computational cost when combined with the wrappers for Undersample/SMOTE and Local Sampling, we restrict our experiments with Undersample/SMOTE and LS with C4.5, which is also the most commonly used classifier with sampling for imbalanced datasets in the literature. Throughout the remainder of this paper, we will abbreviate undersampling/SMOTE as $UnSM$.

The results include $AUROC$ for all five approaches — LS, $UnSM$, COG-OS, C4.5, and SVM. For completeness and repeatability, we also show the parameter values (amount of sampling for $UnSM$ and LS, and number of clusters and replications of the minority class for COG-OS). As will be described later, we use robust statistical measures to evaluate the performance and difference among the different methods.

5.3 Datasets

Table 2 describes the characteristics of the datasets used in our experimental analysis of local sampling. Several datasets were acquired from various real-world scenarios originally considered in [6]. E-State contains electrotopo-

Table 3. $AUROC$ values averaged over 5x2 folds for all methods, along with standard deviations. Statistical significance is calculated using F-test at 95% confidence using the Alpaydin method. If a classifier performance is in bold, it implies $C4.5 + LS$ significantly improved over that classifier. *Italicized* values indicate the corresponding method significantly improved over $C4.5 + LS$. Note that C4.5+LS achieves a higher performance than all the other methods for 17 of the 20 datasets. In addition, the average rank for each classifier is also listed, showing C4.5+LS achieves the best rank and significantly so (the ✓ shows that C4.5+LS achieves statistically significantly lower ran than the corresponding classifier). We use the Friedman test to compare the ranks at 95% confidence interval [12].

Dataset	$C4.5$	SVM	$C4.5 + COGOS$	$SVM + COGOS$	$C4.5 + UnSM$	$C4.5 + LS$
Boundary	**0.509 ± 0.006**	0.758 ± 0.016	0.667 ± 0.034	0.776 ± 0.004	0.671 ± 0.035	0.694 ± 0.039
Breast-w	0.922 ± 0.014	0.938 ± 0.001	**0.956 ± 0.015**	**0.940 ± 0.008**	0.961 ± 0.018	0.973 ± 0.015
Calmodoulin	**0.675 ± 0.008**	0.724 ± 0.012	0.684 ± 0.011	*0.821 ± 0.008*	**0.686 ± 0.006**	0.698 ± 0.005
Compustat	0.792	0.584	0.844	0.738	0.867	0.870
E-State	**0.555 ± 0.056**	**0.567 ± 0.009**	0.620 ± 0.007	0.623 ± 0.019	0.624 ± 0.006	0.636 ± 0.004
Fourclass	**0.973 ± 0.006**	**0.808 ± 0.008**	0.984 ± 0.002	**0.881 ± 0.019**	**0.978 ± 0.005**	0.988 ± 0.003
German.numer	0.736 ± 0.016	0.753 ± 0.001	0.706 ± 0.024	0.753 ± 0.001	0.744 ± 0.010	0.762 ± 0.007
Intrusion	0.970	0.926	0.982	0.976	0.973	0.988
Letter	**0.964 ± 0.003**	**0.678 ± 0.028**	**0.954 ± 0.002**	**0.857 ± 0.005**	**0.969 ± 0.001**	0.975 ± 0.001
Mammography	**0.887 ± 0.005**	**0.895 ± 0.013**	**0.903 ± 0.008**	0.906 ± 0.015	**0.909 ± 0.012**	0.924 ± 0.008
Oil	**0.757 ± 0.015**	**0.543 ± 0.007**	**0.807 ± 0.014**	**0.690 ± 0.038**	0.807 ± 0.036	0.858 ± 0.026
Page	**0.962 ± 0.001**	**0.930 ± 0.005**	**0.979 ± 0.004**	**0.955 ± 0.001**	**0.983 ± 0.002**	0.986 ± 0.002
Pendigits	0.987 ± 0.002	**0.984 ± 0.003**	0.995 ± 0.001	0.995 ± 0.001	0.993 ± 0.001	0.994 ± 0.002
Phoneme	**0.897 ± 0.003**	**0.799 ± 0.003**	**0.910 ± 0.003**	**0.861 ± 0.002**	**0.913 ± 0.006**	0.920 ± 0.005
PhosS	**0.647 ± 0.026**	**0.692 ± 0.005**	**0.631 ± 0.015**	**0.691 ± 0.013**	**0.695 ± 0.003**	0.710 ± 0.004
Pima	**0.742 ± 0.023**	0.801 ± 0.014	0.779 ± 0.014	0.797 ± 0.006	0.787 ± 0.010	0.818 ± 0.009
Satimage	**0.910 ± 0.010**	**0.690 ± 0.031**	**0.906 ± 0.006**	0.927 ± 0.001	**0.921 ± 0.004**	0.927 ± 0.004
Segment	**0.973 ± 0.007**	**0.977 ± 0.002**	0.992 ± 0.002	**0.980 ± 0.001**	**0.986 ± 0.004**	0.993 ± 0.001
Splice	0.954 ± 0.012	**0.841 ± 0.014**	0.958 ± 0.008	**0.849 ± 0.011**	0.962 ± 0.005	0.976 ± 0.006
SVMguide1	**0.982 ± 0.004**	**0.980 ± 0.001**	**0.990 ± 0.001**	**0.982 ± 0.001**	**0.991 ± 0.000**	0.992 ± 0.001
Avg Rank	4.85	4.90	3.65	3.53	2.75	1.33
Friedman $\alpha = .05$	✓	✓	✓	✓	✓	—

logical state descriptors for a series of compounds from the National Cancer Institute's Yeast AntiCancer drug screen. Mammography is highly imbalanced and records information on calcification in a mammogram. Oil is provided by [17]; it is relatively small and very noisy. The Phoneme dataset originates from the ELENA project and is used to distinguish between nasal and oral sounds. Compustat North America is a financial database of U.S. and Canadian publicly held companies, accumulated since 1995. Intrusion is a subset of data acquired from the KDD'99 Challenge on intrusion detection with **u2r** as the minority class and **normal** as the majority class [16]. Boundary, Calmodoulin, and PhosS are various biological datasets [20]. FourClass, German.Numer, Splice, and SVMGuide1 all are available with $LIBSVM$ [5]. The remaining datasets all originate from the UCI repository [2]. Some of these are originally multiple class datasets and were converted into 2-class problems by keeping the smallest class as minority and the rest as majority. The exception is Letter, for which each vowel became a member of the minority class, set against all of the consonants. Aside from stated modifications, each dataset is used "as is."

5.4 Results

The $AUROC$ values for each dataset are presented in Table 3. When $C4.5 + LS$ significantly improves over a classifier, then that classifier's value is in **bold**. When a classifier's value is in *italics*, it significantly improves over $C4.5 + LS$. From these results, the following conclusions about the performance of each classifier may be drawn.

From Table 3, we note that LS improves $C4.5$ and $C4.5 + UnSM$ categorically. This indicates that identifying local rather than global sampling levels is a very effective strategy. In fact, the $AUROC$ values produced through $C4.5 + LS$ are quite compelling, producing the best result on 17 of 20 datasets, with one result a tie with $SVM + COGOS$ on Satimage. $C4.5 + LS$ is outperformed by both $SVM + COGOS$ and SVM on Boundary and Calmodoulin, datasets for which SVM naturally outperforms $C4.5$ by a wide margin. Both datasets are highly dimensional, and as such SVM naturally lends itself to the high dimensional datasets. Despite the high dimensionality, Local Sampling is still able improve over all other $C4.5$ based methods. For Pendigits, we note that there is not quite

Table 4. Parameters optimizing each classifier. For COG-OS, they are given as (k, r), where k is the number of k-means clusters found for the majority class and r is the number of replications of the minority class, while for Undersample/SMOTE and Local Sampling, they are given as $(Undersample, SMOTE)$.

Dataset	$C4.5 + COGOS$	$SVM + COGOS$	$C4.5 + UnSM$	$C4.5 + LS$
Boundary	(4.6, 8.0)	(4.6, 2.7)	(28, 325)	(28, 240)
Breast-w	(4.8, 2.9)	(4.6, 6.7)	(64, 180)	(54, 217)
Calmodoulin	(3.4, 6.7)	(2.4, 3.7)	(35, 285)	(42, 155)
Compustat	(2, 8)	(2, 7)	(30, 50)	(34, 79)
E-State	(4.8, 7.4)	(6.6, 7.0)	(38, 255)	(36, 247)
Fourclass	(3.8, 2.1)	(6.6, 6.2)	(55, 285)	(41, 231)
German.numer	(3.2, 3.9)	(2.0, 6.0)	(48, 280)	(56, 208)
Intrusion	(6, 2)	(8, 10)	(30, 300)	(43, 244)
Letter	(6.8, 3.3)	(8.0, 3.4)	(60, 205)	(49, 170)
Mammography	(5.0, 1.0)	(4.0, 5.5)	(27, 330)	(37, 195)
Oil	(4.0, 2.5)	(4.0, 4.0)	(47, 300)	(42, 151)
Page	(2.0, 4.5)	(3.0, 6.4)	(50, 290)	(49, 234)
Pendigits	(5.2, 3.3)	(8.0, 2.3)	(59, 280)	(56, 256)
Phoneme	(5.8, 6.6)	(3.6, 1.4)	(56, 250)	(56, 170)
PhosS	(4.0, 7.2)	(2.8, 1.9)	(30, 160)	(29, 205)
Pima	(4.4, 8.1)	(2.0, 3.4)	(32, 260)	(57, 211)
Satimage	(2.4, 8.7)	(7.0, 2.8)	(44, 345)	(40, 397)
Segment	(7.4, 4.4)	(6.2, 6.3)	(45, 250)	(37, 170)
Splice	(2.6, 3.3)	(5.2, 2.0)	(73, 185)	(70, 84)
SVMguide1	(2.0, 5.9)	(5.2, 2.0)	(43, 280)	(42, 166)

as large a discrepancy in base classifier performance. We note that COG-OS improves both $C4.5$ and SVM above Local Sampling. Since it does so for both classifiers, we presume that COG-OS is quite effective in extracting a subclass for each digit forming the majority class, which is more effective to improve $C4.5$ than injecting synthetic examples. Overall, on the 18 datasets for which significance is tested, there is a significant improvement on 14 datasets when comparing to $C4.5$, 13 when comparing to SVM, 8 when comparing to $C4.5 + COGOS$, 10 when comparing to $SVM + COGOS$, and 10 when comparing to $C4.5 + UnSM$. Considering the conservative nature of the 5x2 cv, these results are noteworthy. *Finally, when we compare the performance ranking of different methods, $C4.5 + LS$ emerges as a clear winner, at 95% confidence per the Friedman tests using the Holm procedure*. This is a compelling result, clearly establishing the performance improvements achieved by local sampling.

We also point out that dividing the majority class into sub-classes through COG-OS is effective in improving the $AUROC$ values over the base classifier. While it often underperforms, as compared to $C4.5 + LS$ (and statistically significantly so), it improves the $AUROC$ of $C4.5$ on 16 of 20 datasets and SVM on 18. COG-OS, using k-means clustering, is able to effectively exploit the inherent structures in the majority class space to improve performance over the base classifier. However, we show that it is more impactful to decompose the minority class space using a supervised partitioning method.

Table 4 shows the sampling parameters producing optimal results. For COG-OS, the k value selected for $C4.5$ and SVM are moderately similar, while r values are quite divergent. The Pearson correlation comparison for k values between $C4.5$ and SVM is 0.46, showing moderate correlation; however, r values are weakly negatively correlated at -0.35. $C4.5$ and SVM will tend to select r values at opposite ends of the spectrum. Fisher's z-test suggests that these two correlations are significantly different. We note that the undersamplings produced by $UnSM$ and LS are more similar with a Pearson of 0.74 than the determined SMOTE levels, which correlate at 0.61. However, both sampling levels are highly correlated and Fisher's z-test suggests that these correlations are similar with 95% confidence. Therefore, we conclude the strength of Local Sampling derives from its ability to manipulate specific regions of samplings. These parameters will allow for the **reproducibility** of our results. *We will also readily provide all the real-world datasets and source code to the interested readers.*

6 Conclusions

There are various key conclusions to be drawn from this work. We elucidate the key observations on the behavior of classifiers in the presence of class imbalance, which also helped develop the thesis for the proposed framework of Local Sampling (LS). We observed how properties of the data, aside from class skew itself, affect end classifier performance and that different sampling levels may produce

optimal performances on similar datasets, even when the class skew is identical. We also noted that a classifier improved through global sampling levels may be insensitive to the peculiarities of different components or modalities in the data, resulting in a suboptimal performance. We demonstrated that by first discovering components or segments in data, and then applying sampling locally to each of the segments, resulted in the best global performance — *starting globally, optimizing locally, and predicting globally*.

The main contribution of the paper general framework for Local Sampling (LS) and a class skew insensitive partitioning method, effective in this application. This framework is able to achieve significantly better performances than other contemporary methods. As per the Friedman test, it achieves best performance, and statistically so, across 20 datasets derived from different domains.

References

[1] E. Alpaydin. Combined 5x2cv F Test for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 11(8):1885–1892, 1999.

[2] A. Asuncion and D. Newman. UCI Machine Learning Repository, 2007.

[3] G. Batista, R. Prati, and M. Monard. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explorations*, 6(1):20–29, 2004.

[4] S. Bickel, M. Brückner, and T. Scheffer. Discriminative Learning for Differeing Training and Test Distributions. In *ICML*, pages 81–88, 2007.

[5] C. Chang and C. Lin. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[6] 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, 2002.

[7] N. V. Chawla, D. A. Cieslak, L. O. Hall, and A. Joshi. Automatically countering imbalance and its empirical relationship to cost. *Utility-Based Data Mining: A Special issue of the International Journal Data Mining and Knowledge Discovery*, 2008.

[8] N. V. Chawla, N. Japkowicz, and A. Kolcz. Editorial: Learning from Imbalanced Datasets. *SIGKDD Explorations*, 6(1):1–6, 2004.

[9] D. A. Cieslak and N. V. Chawla. Detecting fractures in classifier performance. In *IEEE International Conference on Data Mining (ICDM)*, 2007.

[10] D. A. Cieslak and N. V. Chawla. A Framework for Monitoring Classifiers' Performance: When and Why Failure Occurs? *Knowledge and Information Systems*, 2008.

[11] D. A. Cieslak and N. V. Chawla. Learning Decision Trees for Unbalanced Data. In *European Conference on Machine Learning (ECML)*, pages 241–256, 2008.

[12] J. Demsar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[13] T. G. Dietterich. Approximate statistical test for comparing supervised classiffication learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

[14] C. Elkan. The Foundations of Cost-Sensitive Learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 973–978, 2001.

[15] N. Japkowicz. Class Imbalance Problem: Significance & Strategies. In *International Conference on Artificial Intelligence (ICAI)*, pages 111–117, 2000.

[16] The Third International Knowledge Discovery and Data Mining Tools Competition, 1999.

[17] M. Kubat, R. C. Holte, and S. Matwin. Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*, 30:195–215, 1998.

[18] F. Provost and P. Domingos. Tree Induction for Probability-Based Ranking. *Machine Learning*, 52(3):199–215, September 2003.

[19] J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1:81–106, 1986.

[20] P. Radivojac, N. V. Chawla, A. K. Dunker, and Z. Obradovic. Classification and knowledge discovery in protein databases. *Journal of Biomedical Informatics*, 37:224–239, 2004.

[21] C. Rao. A Review of Canonical Coordinates and an Alternative to Corresponence Analysis using Hellinger Distance. *Questiio (Quaderns d'Estadistica i Investigacio Operativa)*, 19:23–63, 1995.

[22] G. Weiss. Mining with Rarity: A Unifying Framework. *ACM SIGKDD Explorations*, 6:7–19, 2004.

[23] J. Wu, H. Xiong, P. Wu, and J. Chen. Local Decomposition for Rare Class Analysis. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 814–823, 2007.