MixBoost: Synthetic Oversampling using Boosted Mixup for Handling Extreme Imbalance

Anubha Kabra et al

Media and Data Science Research Lab, Adobe

ICDM 2020

Park, MinKyu

2021.08.12

Dongguk University

Artificial Intelligence Laboratory

* 2020 IEEE International Conference on Data Mining (ICDM).

Motivation

- In most of datasets, the instances of one class (the majority class) far outnumber those of the other class (the minority class) is a challenging problem.
- Data augmentation methods are used to alleviate this problem.
- These methods generate good quality synthetic instances in regions of the input space where the classification model is already accurate.
- Existing methods generate synthetic **homogeneous** instances, i.e., instances that belong to a single class (usually the minority).
- Recent work in the domain of Computer Vision has demonstrated effect of **non-homogeneous hybrid samples** to learn robust representation.

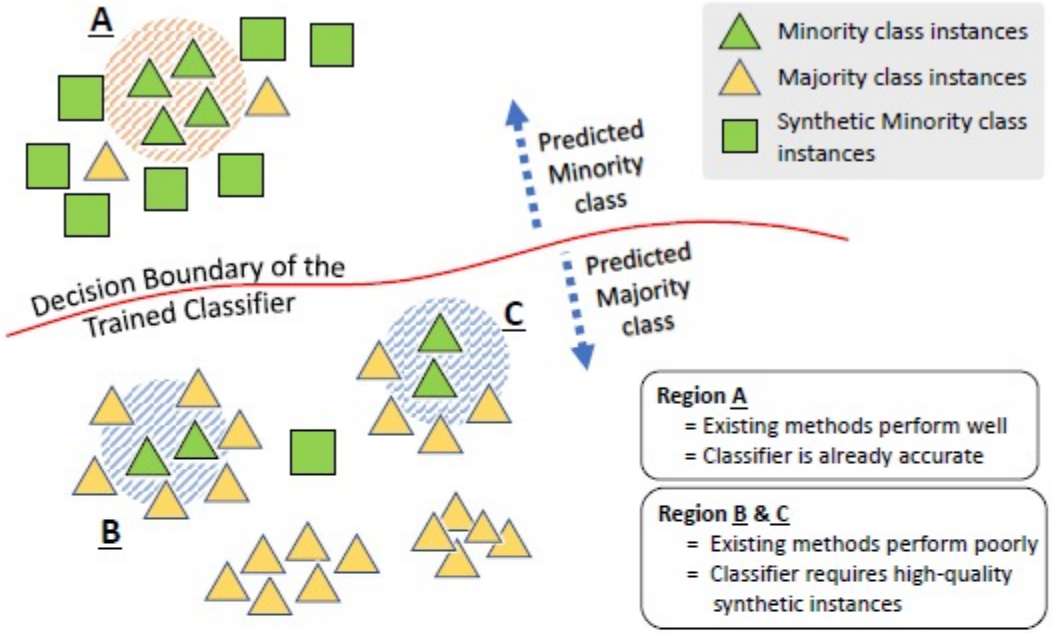


Fig. 1: Illustration to describe the limitation of synthetic oversampling techniques. Existing methods (SMOTE [5] and its variants, SWIM [18]) select candidate instances from one of the classes and create synthetic instances based on these selected instances. Therefore, most generated synthetic instances lie near clusters (often within the convex hull) of instances that are often already correctly classified by the classification model (region A). Further, these methods generate fewer and poorer quality synthetic instances in regions of the input space where the model does not perform well (regions B and C).

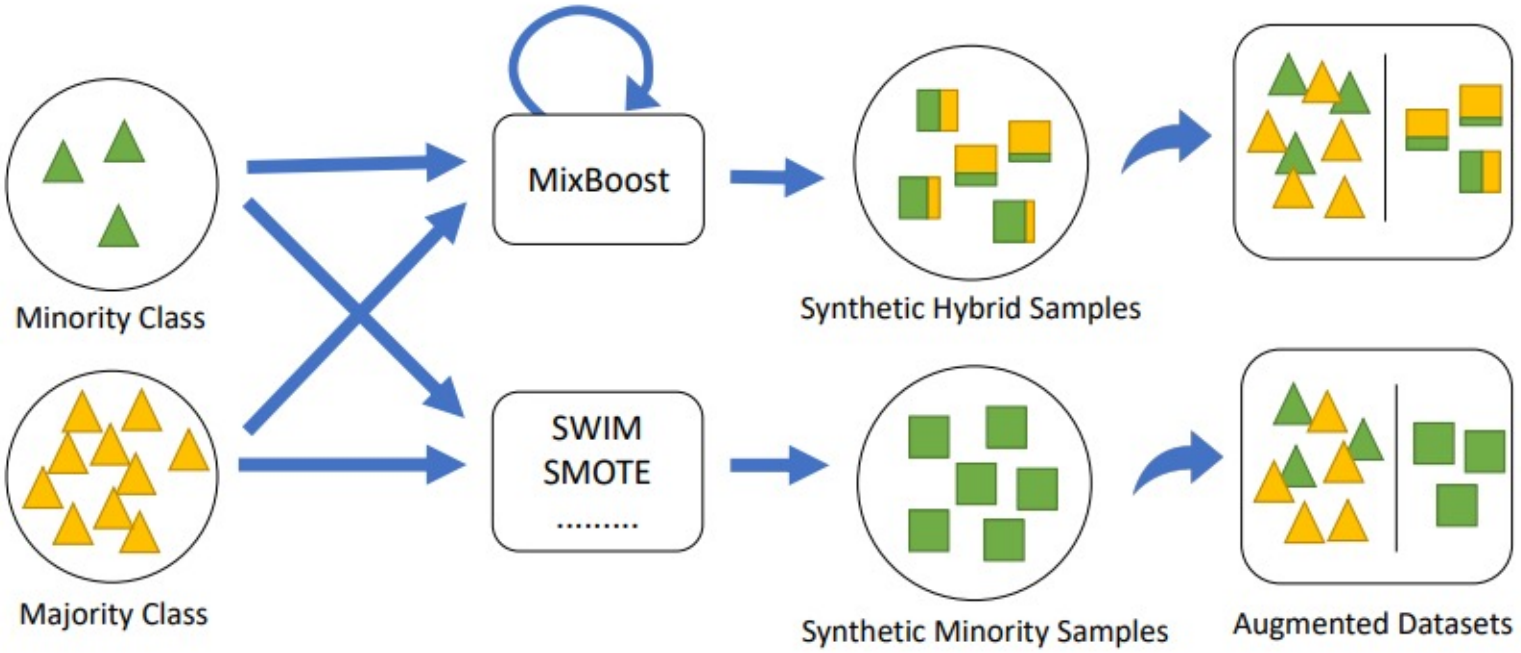
methods to augment an imbalanced dataset

- Under-sampling methods :
  - discard instances of the majority class at random to balance the class distribution → a loss of information

- Over-sampling methods :
  - duplicate instances of the minority class at random to balance the class distribution
  - **SMOTE** [5] creates synthetic minority instances by interpolating minority class instances in the training data → SMOTE ignores majority class data
  - To expand the space spanned by generated instances, **SWIM** [18] creates instances by inflating minority class data along the density contours of majority class data

Contribution

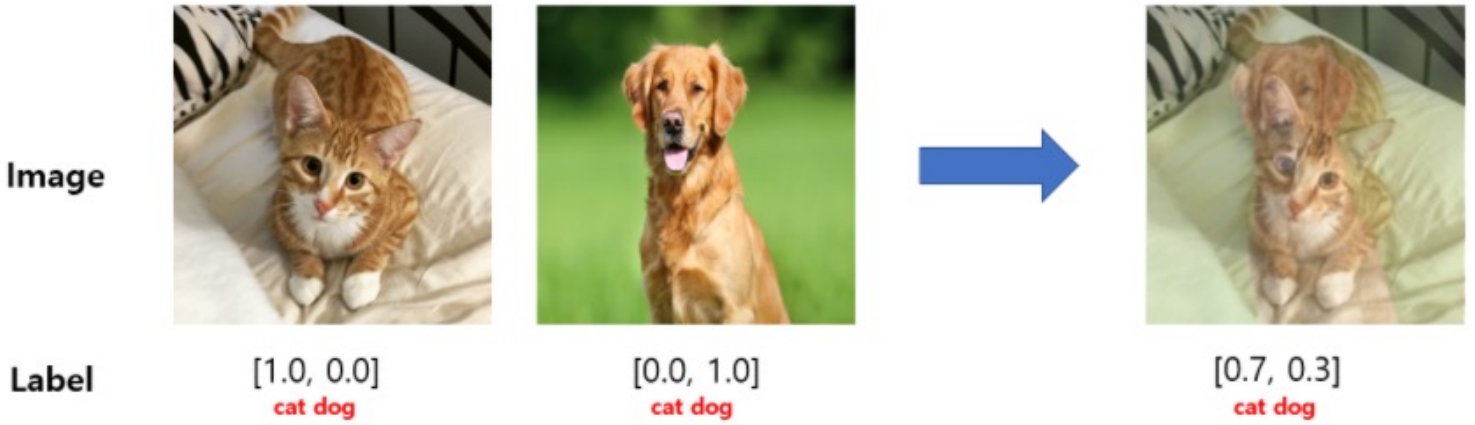
- To learn robust representations, Generate non-homogeneous hybrid samples that have elements of majority and minority class.
- Key idea
  - First, Select the instances for mixing intelligently.
  - Second, Mix instances of the minority and majority class to generate synthetic hybrid instances that have elements of both classes.

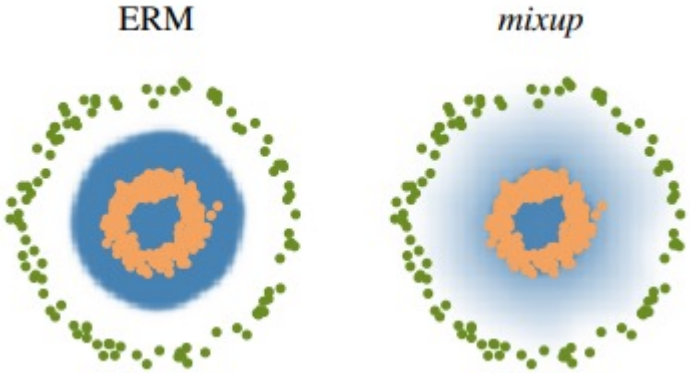


Mixup: Example

$$\begin{aligned} \hat{x} &= \lambda x_i + (1-\lambda)x_j, \\ \hat{y} &= \lambda y_i + (1-\lambda)y_j, \end{aligned}$$

where $\lambda \in [0, 1]$ is a random number





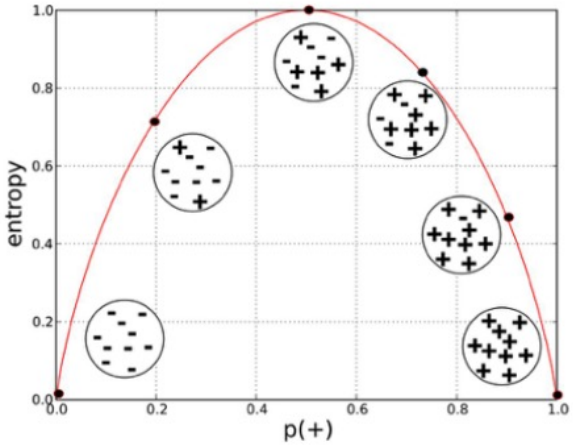
MixBoost

- generates synthetic hybrid instances by interpolating instances from the majority and minority classes

1. Candidate Selection (Boost): We sample candidate instances from the majority and minority classes in D_{train}^{orig} prior to mixing.

2. Hybrid Generation (Mix): We mix the sampled instances to generate synthetic hybrid instances D^{hyb} := (X^{hyb}; Y^{hyb}) denotes the set of synthetic instances generated in this step (and each synthetic instance is correspondingly referred to as d^{hyb}).

Candidate Selection (the Boost step)



- **<u>Two alternative strategies</u>**
- **R-Selection**
  - Randomly selects (majority and minority) candidates.

$$p_0^i = 1/n_0 \ \forall i, \qquad p_1^j = 1/n_1 \ \forall j$$

- **Entropy Weighted (EW) Selection:**
  - Actively weighting candidates with the uncertainty
  - $\vec{y}_{pred}^{\,i}$ is the probability distribution over target classed output by the classifier M

$$E^i = Entropy(\vec{y}_{pred}^{\,i}) = -\sum y_{pred}^i * log(y_{pred}^i) \qquad (2)$$

  - E^i measures the distance of the instance to the decision boundary of the classifier.
  - A high E^i implies that M is uncertain about the ground-truth class for the instance

➔ the instance is close to decision boundary.

  - A low E^i implies that M is certain about the ground-truth class for the instance

➔the instance is far from to decision boundary

Candidate Selection (the Boost step)

- **Entropy Weighted (EW) Selection:**
  - augmenting training dataset with synthetic instances in vicinity of high entropy feature sub-spaces can improve model training performance
  - $\hat{E}_0 = \Sigma\, E^i \forall x_0^i \left(x^i \; with \; class \; c_0\right)$: the sum of entropy values for majority class instances, $\hat{E}_1 = \Sigma\, E^i \forall x_1^i \left(x^i \; with \; class \; c_1\right)$: the sum of entropy values for minority class instances
  - $P\left(x^i \,|c_0\right)$ and $P\left(x^i \,|c_0\right)$ denote the entropy ratios

$$P(x^i|c_0) = \frac{E^i}{\hat{E}_0}, \qquad P(x^j|c_1) = \frac{E^j}{\hat{E}_1}$$

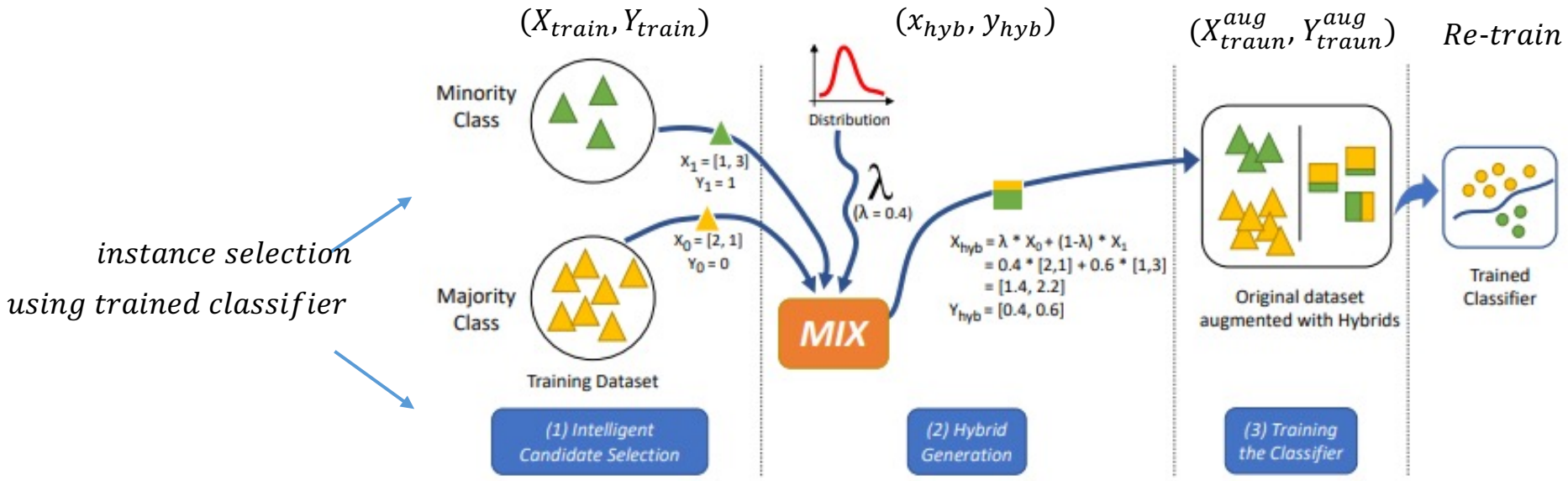
low-high **selection**

Hybrid Generation (the Mix step)

- (x_0, y_0) and (x_1, y_1) is instances selected (in the Boost step) from majority and minority class respectively

$$x_{hyb} = \lambda x_0 + (1 - \lambda) x_1 \qquad y_{hyb} = \lambda y_0 + (1 - \lambda) y_1$$

- The classifier is re-trained with the original data augmented with the hybrid instances prior to the next iteration of MixBoost.



Experimental Settings

- Datasets
  - train : test = 5 : 5
  - Randomly down-sample to simulate different levels of extreme imbalance
  - ➔Test at three levels of imbalance, size 4, 7, and 10.
- Evaluation:
  - $g - mean = \sqrt{TPR \times TNR}$
  - ROC-AUC scores.

Dataset	Features	No. of majority instances	R4	R7	R10
Abalone 9-18	8	689	1:173	1:99	1:69
Diabetes	8	500	1:125	1:72	1:50
Wisconsin	9	444	1:111	1:64	1:456
Wine Q. Red 4	11	1546	1:387	1:221	1:155
Wine Q. White	11	880	1:220	1:126	1:88
Vowel 10	13	898	1:225	1:129	1:90
Pima Indians	8	500	1:125	1:72	1:50
Vehicle 0	18	641	1:160	1:91	1:64
Vehicle 1	18	624	1:156	1:89	1:62
Vehicle 2	18	622	1:155	1:88	1:62
Vehicle 3	18	627	1:156	1:89	1:62
Ring Norm	20	3736	1:934	1:534	1:374
Waveform	21	600	1:150	1:86	1:60
PC4	37	1280	1:320	1:183	1:128
Piechart	37	644	1:161	1:92	1:65
Pizza Cutter	37	609	1:153	1:87	1:61
Ada Agnostic	48	3430	1:858	1:490	1:343
Forest Cover	54	2970	1:743	1:425	1:297
Spam Base	57	2788	1:697	1:399	1:279
Mfeat Karhu.	64	1800	1:450	1:258	1:180

TABLE I: Description of the datasets used in our experiments. To ensure evaluation consistency, we use the same datasets and configuration as proposed by [18]. R4, R7, and R10 denote the ratio of class imbalance (minority:majority) after down-sampling the training datasets to have 4, 7, and 10 minority class instances respectively to simulate the extreme imbalance [18] scenarios (as discussed in Section I)

Results

Dataset	Baseline	ALT	SWIM	*MixBoost*
Abalone 9-18	0.481	0.612	0.723	**0.743**
Diabetes	0.259	0.509	0.509	**0.701**
Wisconsin	0.874	0.956	0.958	**0.969**
Wine Q. Red 4	0.224	0.502	0.535	**0.815**
Wine Q.White 3v7	0.451	0.572	0.730	**0.750**
Vowel 10	0.724	0.738	0.812	**0.845**
Pima Indians	0.276	0.479	0.509	**0.700**
Vehicle 0	0.534	0.758	0.814	**0.900**
Vehicle 1	0.541	0.739	**0.791**	0.735
Vehicle 2	0.450	0.549	0.560	**0.880**
Vehicle 3	0.402	0.505	0.569	**0.651**
Ring Norm	0.274	**0.933**	0.899	0.580
Waveform	0.301	0.701	0.688	**0.844**
PC4	0.572	0.559	0.611	**0.737**
PieChart	0.455	0.516	0.576	**0.741**
Pizza Cutter	0.468	0.506	0.552	**0.725**
Ada Agnostic	0.451	0.445	0.539	**0.690**
Forest Cover	0.561	0.554	0.550	**0.917**
Spam Base	0.440	0.550	0.685	**0.872**
Mfeat Karhunen	0.274	**0.933**	0.899	0.927

TABLE II: Comparative *g-mean* results (mean) for *MixBoost* with existing over-sampling methods. These results represent the R4 setting where the training dataset has 4 minority class instances. Baseline refers to the case where classifier is trained without data augmentation. ALT is the score of the best performing data augmentation strategy (other than SWIM [18] and *MixBoost*) as described in Section III-D. The best score

Dataset	*MixBoost*	
	R-Selection	EW-Selection
Abalone 9-18	0.743 ± 0.03	0.735 ± 0.06
Diabetes	0.701 ± 0.05	0.560 ± 0.04
Wisconsin	0.960 ± 0.02	0.969 ± 0.08
Wine Q. Red 4	0.714 ± 0.04	0.815 ± 0.08
Wine Q. White 3v7	0.743 ± 0.06	0.750 ± 0.05
Vowel 10	0.845 ± 0.04	0.854 ± 0.07
Pima Indians	0.700 ± 0.05	0.597 ± 0.04
Vehicle 0	0.900 ± 0.05	0.850 ± 0.02
Vehicle 1	0.700 ± 0.03	0.735 ± 0.03
Vehicle 2	0.880 ± 0.02	0.638 ± 0.03
Vehicle 3	0.651 ± 0.06	0.600 ± 0.03
Ring Norm	0.550 ± 0.04	0.580 ± 0.03
Waveform	0.812 ± 0.03	0.844 ± 0.05
PC4	0.720 ± 0.08	0.737 ± 0.04
PieChart	0.611 ± 0.06	0.741 ± 0.07
Pizza Cutter	0.725 ± 0.05	0.678 ± 0.07
Ada Agnostic	0.690 ± 0.02	0.648 ± 0.03
Forest Cover	0.910 ± 0.05	0.917 ± 0.02
Spam Base	0.872 ± 0.03	0.834 ± 0.05
Mfeat Karhunen	0.888 ± 0.07	0.927 ± 0.05

TABLE III: Comparative *g-mean* results (mean and standard deviation for 30 independent runs) for the different candidate selection strategies (refer to Section IV for details) of our approach *MixBoost*.

- MixBoost using R-selection or EW-selection outperforms existing methods on 18 out of 20 datasets.
- EW-selection is significantly better than R-selection on several datasets.

* ALT: ROS, RUS, SMOTE,B1, B2, SMOTE with Tomek Links, ADASYN

Ablation Study

A. Impact of sampling instances over multiple iterations

Dataset	*MixBoost-1-Iter*	*MixBoost*
Pima Indians	0.491 ± 0.02	**0.597 ± 0.04**
Waveform	0.843 ± 0.04	**0.844 ± 0.05**
PC4	0.613 ± 0.08	**0.737 ± 0.04**
Piechart	0.721 ± 0.02	**0.741 ± 0.07**
Forest Cover	0.910 ± 0.00	**0.917 ± 0.02**

TABLE V: *g-mean* scores for the single step (*MixBoost*-1-Iter) and the proposed *MixBoost*. For all selected datasets, the iterative variant of *MixBoost* outperforms the single step one.

B. Impact of choice of distributions for sampling λ

Dataset	Uniform	Beta
Pima Indians	0.389 ± 0.08	**0.597 ± 0.04**
Waveform	0.661 ± 0.01	**0.844 ± 0.05**
PC4	0.242 ± 0.01	**0.737 ± 0.04**
Piechart	0.312 ± 0.07	**0.741 ± 0.07**
Forest Cover	0.629 ± 0.01	**0.917 ± 0.02**

TABLE VI: *g-mean* scores when we sample λ from different distributions. For all datasets, sampling λ from the *Beta(0.5,0.5)* distribution leads to the best performance.

E. Impact of number of generated synthetic hybrid-instances

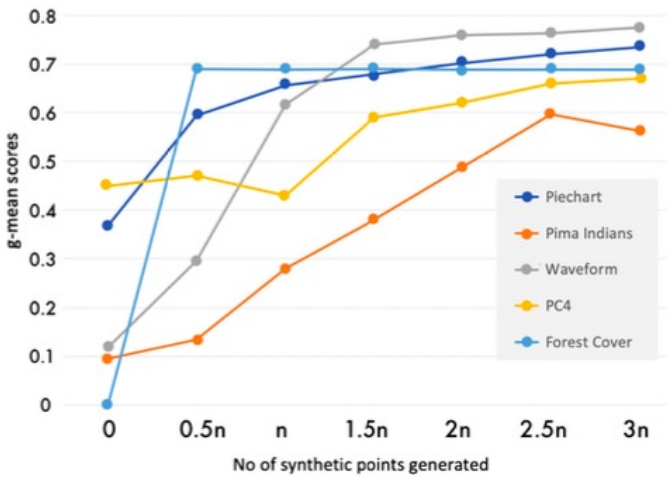


Fig. 6: Variation in *g-mean* scores (averaged over 30 runs) for *MixBoost* as we increase the number of generated synthetic hybrid instances. n represents the total number of training instances in the original dataset. The different color lines indicate different datasets. The gain of generating an additional $0.5n$ synthetic instances is initially high and then falls gradually as the number of generated instances increases.

Related Work

- classification problems on imbalanced datasets
  - First, sampling-based approaches and second, cost-based approaches
  - focus on sampling-based approaches
    - The most straightforward re-sampling strategies are Random under Sampling (**RUS**), and Random over Sampling (**ROS**)
    - SMOTE (Synthetic Minority Oversampling Technique) : generates a synthetic instance by interpolating the k-nearest neighbors of a minority class instance in the training data
    - Extensions of SMOTE [9] add **a post-processing step** that tries to remove generated instances that might degrade the performance of the classifier.
      → Adaptive Synthetic Oversampling (ADASYN) [14], borderline SMOTE [12], Majority Weighted Minority Oversampling [13]
    - SWIM[18] uses information from the majority class to generate synthetic instances
    - use a Generative Adversarial Network (GAN) trained on minority class data to generate synthetic training instances

Conclusion

- we tackle the problem of binary classification on extremely imbalanced datasets.
- we propose *MixBoost*, a technique for synthetic iterative over-sampling. *MixBoost* intelligently selects and then combines instances from the majority and minority classes to generate synthetic hybrid instance.
- future study
  - focus on evaluating *MixBoost* for multi-class classification
  - adapt the idea of iterative sampling
  - generation through interleaved Mix and Boost steps for regression tasks

감사합니다.