Trainable Undersampling for Class-imbalanced Learning

Minlong Peng

Computer Science and Technology, Fudan University

Problem of Class-imbalance Learning

- In class-imbalance tasks, the evaluation metrics are often not the “accuracy”.
- Training object of traditional supervised learning algorithms, or formally their loss functions, are only consistent with the accuracy metric.

Typical Solutions

Model Based:

Design a loss function that is consistent with the evaluation metric of a given task.

Pros.:

1. Apply to different tasks with the same evaluation metric.
2. Usually outperform other methods.

Cons.:

1. Evaluation metric specific.
2. Hard to design the loss function.
3. Does not apply to non-parametric models, such as KNN.

Typical Solutions

Data Based:

Change the input distribution, making the training object of the classifier consistent with the evaluation metric on the original input distribution.

Pros.:

1. Easy to implement (e.g., undersampling, oversampling).
2. Apply to non–parametric models.

Cons.:

1. Hard to obtain the optimum solution, since data sampling is often heuristic–based.

Motivation of this Work

Connect data sampling with the used classifier and evaluation metric.

Proposed Solution

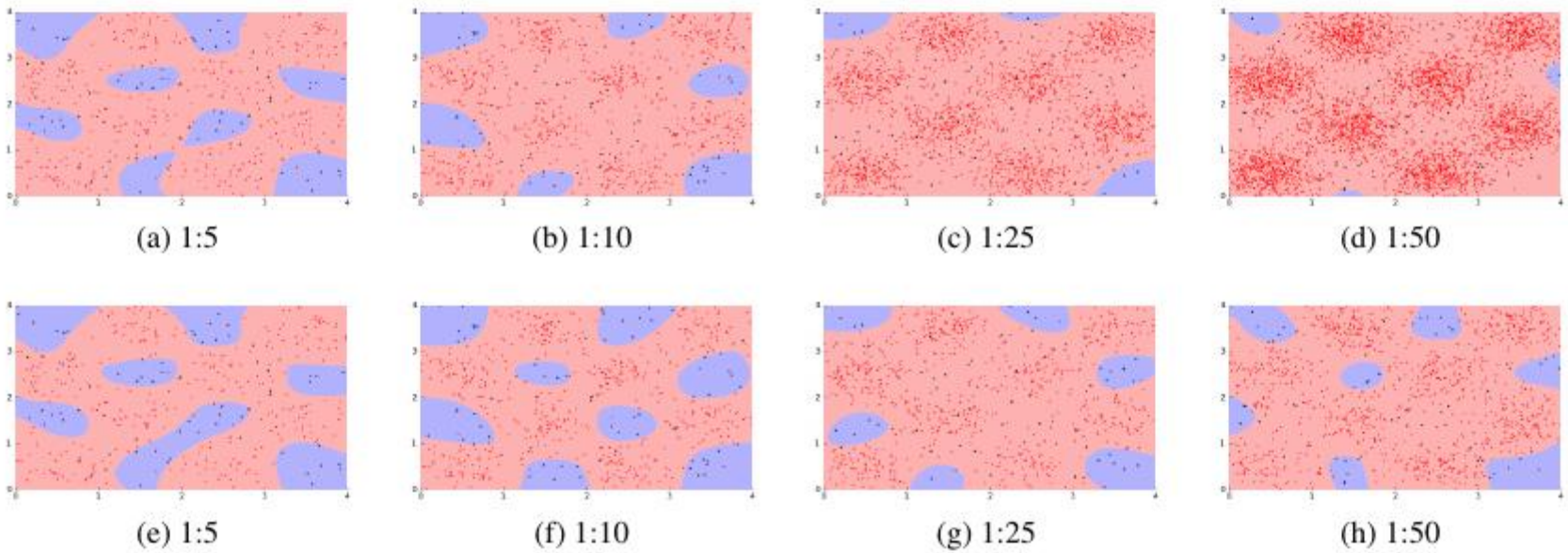
- Find a subset of the training set that maximize the evaluation metric of the task using a given classifier.

$$\boldsymbol{w}^*(\{\mathbf{X},\mathbf{Y}\}) := \underset{\Im(\{\mathbf{X},\mathbf{Y}\})\subseteq\{\mathbf{X},\mathbf{Y}\}}{\arg\max} \boldsymbol{r}\left(\{\mathbf{Y}, \boldsymbol{f}(\mathbf{X}; \Im(\{\mathbf{X},\mathbf{Y}\}))\}\right)$$

Proposed Solution

1. Choose a subset of the training dataset using policy network
2. Train the model with the chosen subset
3. Evaluate model performance on the original training dataset and obtain a reward (performance on the given measurement)
4. update the policy network with the obtained reward.
5. return to step 1.

Results



(a) 1:5 (b) 1:10 (c) 1:25 (d) 1:50

(e) 1:5 (f) 1:10 (g) 1:25 (h) 1:50

Results

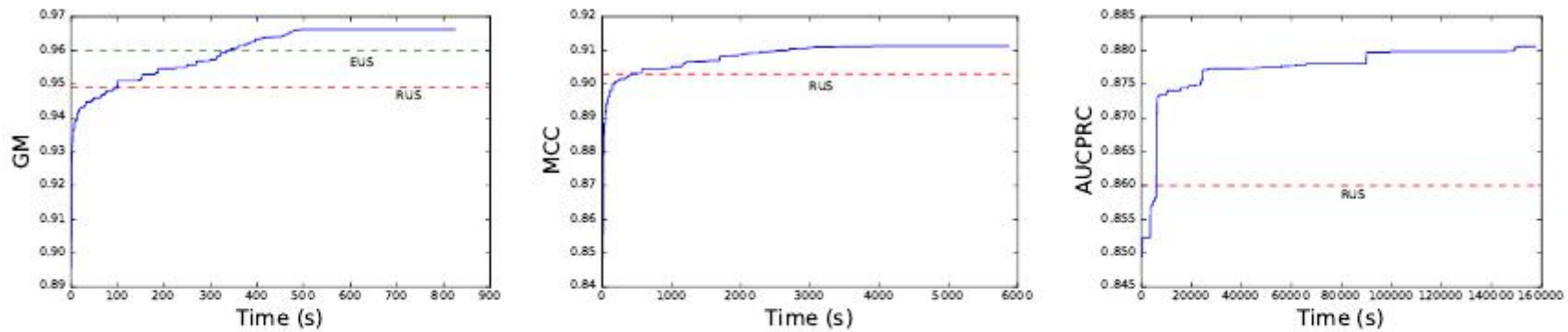
Dataset	#Attribute	#Example	Feature Format	Minority Ratio	Evaluation Metric	Used Classifier
Vehicle	18	846	Numeric	25.65%	GM	SVM (rbf)
Page blocks	10	5,472	Numeric	10.21%	MCC	MLP
Credit Fraud	28	284,807	Numeric	0.17%	AUCPRC	DT
SMS Spam	8,749	5,574	Text	13.41%	$F_{0.5}$	LR
DR	262,144	17,563	Image	26.52%	AUCROC	CNN

Task	ORG	RUS	NearMiss	Cluster	TomekLink	ALLKNN	SMOTE	ADASYN	TU
Vehicle	0.935	0.949	0.877	0.937	0.938	0.858	0.935	**0.964**	**0.964**
Page-blocks	0.897	0.903	0.878	0.877	0.895	0.867	0.897	0.902	**0.915**
Credit Fraud	0.849	0.860	0.817	0.584	0.840	0.809	0.849	0.848	**0.880**
SMS Spam	0.936	0.938	0.931	0.932	0.935	0.933	0.936	0.936	**0.967**
DR	0.930	0.942	0.921	0.933	0.934	0.927	0.930	0.944	**0.958**

Results

Task	ORG	SMOTE	ADASYN	TU	SMOTE+TU	ADASYN+TU
Vehicle	0.935	0.964	0.964	0.964	0.964	**0.965**
Page-blocks	0.897	0.902	0.898	0.915	**0.917**	0.915
Credit Fraud	0.849	0.848	0.849	0.880	**0.881**	0.880
SMS Spam	0.936	0.936	0.936	**0.967**	0.965	**0.967**
DR	0.930	0.944	0.943	**0.958**	0.957	0.956

Results



(a) Vehicle (b) Page blocks (c) Credit Fraud

Figure 2: Time complexity of the proposed method on three tested datasets. The red dot line denotes the performance of the random undersampling method.

Thanks!