Trainable Undersampling for Class-Imbalance Learning

Minlong Peng[1], Qi Zhang[1], Xiaoyu Xing[1], Tao Gui[1], Xuanjing Huang[1]
Yu-Gang Jiang[1], Keyu Ding[2], Zhigang Chen[2]
[1]{mlpeng16, qz, xyxing14, tgui16, xjhuang, ygj}@fudan.edu.cn
[2]{kyding, zgcheng}@iflytek.com

Introduction

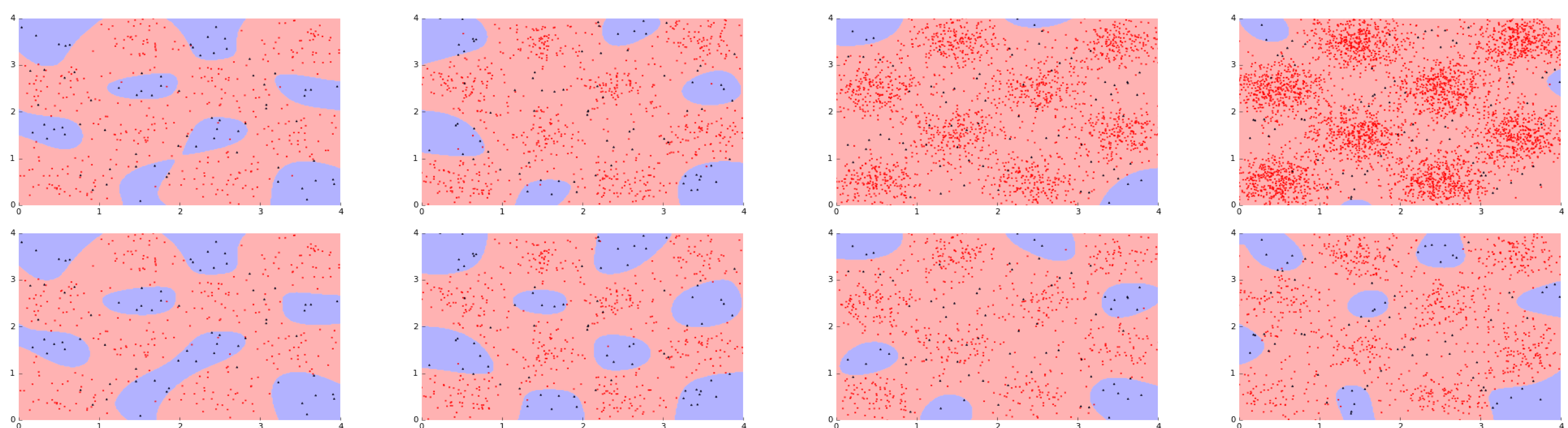
The key idea of the proposed method is to parametrize the data sampler and train it to optimize the classification performance over the evaluation metric. We solve the non-differentiable optimization problem for training the data sampler via reinforcement learning. By incorporating evaluation metric optimization into the data sampling process, the proposed method can learn which instance should be discarded for the given classifier and evaluation metric. As a data level operation, this method can be easily applied to arbitrary evaluation metric and classifier, including non-parametric ones (e.g., C4.5 and KNN). Experimental results on both synthetic and realistic datasets demonstrate the effectiveness of the proposed method. Source code is availabel at https://github.com/v-mipeng/Class-Imbalance-Leawrning.

Methods

Algorithm 1 Trainable Undersampling

Input: training dataset $\{\mathbf{X}, \mathbf{Y}\}$, classification procedure $\boldsymbol{f}$, initial policy $\pi(\boldsymbol{\theta}_0)$, maximum number of iteration N
Initialize: $\pi(\boldsymbol{\theta}) \leftarrow \pi(\boldsymbol{\theta}_0)$; $T \leftarrow$ dataset size $|\{\mathbf{X}, \mathbf{Y}\}|$
repeat
  $V(\boldsymbol{s}) \leftarrow \emptyset$
  for $t = 1$ to T **do**
    $\boldsymbol{s}_t \leftarrow V(\boldsymbol{s}) \cup \{(\mathbf{x}, \mathbf{y})\}$
    choose action $a_t \in \{(\mathbf{x}_t, \mathbf{y}_t), \emptyset\}$ in probability $a_t \sim \pi(a|\boldsymbol{s}_t; \boldsymbol{\theta})$.
    $V(\boldsymbol{s}) \leftarrow V(\boldsymbol{s}) \cup \{a_t\}$
  train the classifier $f(\cdot|V(\boldsymbol{s}))$ on $V(\boldsymbol{s})$
  obtain the reward $R_\tau \leftarrow \mathcal{G}(\{\mathbf{Y}, \boldsymbol{f}(\mathbf{X}; V(\boldsymbol{s}))\})$
  update $\boldsymbol{\theta}$ in the direction that maximizes the reward $\Delta\boldsymbol{\theta} \propto \sum_{t=1}^{T} \frac{\partial \log \pi(a_t|\boldsymbol{s}_t;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} R_\tau$.
until $\pi(\boldsymbol{\theta})$ converges or maximum number of iterations N exceeds.
generate $\boldsymbol{w}(\{\mathbf{X}, \mathbf{Y}\})$ according to Eq. 9.
train $\boldsymbol{f}$ on $\boldsymbol{w}(\{\mathbf{X}, \mathbf{Y}\})$
Return: $\boldsymbol{f}$

Synthetic



Real-world

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**