Noname manuscript No.
(will be inserted by the editor)

Imbalanced classification via robust optimization

Dimitris Bertsimas · Yuchen Wang

Received: DD Month YEAR / Accepted: DD Month YEAR

Abstract Classification on imbalanced datasets is usually a challenging task in machine learning. There are already several methods to solve this problem, but they either delete some data or generate some data artificially. In this paper, we revisit the imbalanced classification problem from a Robust Optimization (RO) view. We propose an algorithm to utilize RO to train Logistic Regression (LR) and Support Vector Machines (SVM) on imbalanced datasets. We show that the proposed method provides a significant performance edge for the F1-metric ($\sim 6\% - 8\%$), a small performance edge on the AUC-metric ($\sim 1\%$) over the state-of-the-art oversampling and undersampling methods on 20 real-world imbalanced datasets, while its running time is higher, but still within practical range (the proposed methods runs in minutes for $n \sim 60,000$).

Keywords Robust Optimization · Imbalanced datasets · Classification

1 Introduction

For classification problems, it is often the case that the number of samples of a given class is much smaller than the number of samples in other classes. This imbalance leads to the so-called imbalanced classification problem, one of the top 10 problems in data mining (Yang and Wu (2006)). For a binary classification problem, the class with more data is called the majority class, and the class with less data is called the minority class (Chawla et al. (2003)). The

Dimitris Bertsimas
Operations Research Center and Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02139, USA E-mail: dbertsim@mit.edu

Yuchen Wang
Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139, USA E-mail: yuchenw@mit.edu

ratio between the majority class and the minority class is called the Imbalance Ratio (IR) (Chawla et al. (2002)). The imbalance classification problem occurs in a variety of areas. For example, the IR is 112 for children with clinically important traumatic brain injury (ciTBI) (Bertsimas, Dunn, Steele, Trikalinos and Wang (2019)) and about 100 for fraud detection (Provost and Fawcett (2001)). In the presence of very imbalanced datasets, the accuracy of the classification methods suffers. Therefore, we need methods to improve the accuracy in such settings.

There are already several methods proposed to solve this problem. The first type of method is to increase the weights of the minority class in the loss function. For example, weighted LR is to set the weight of the minority class to be the IR. By emphasizing the minority class more in this way, the model will predict more accurately on the minority class.

Oversampling is the second type of method to generate more minority class samples in order to convert an imbalanced dataset to a more balanced one. Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al. (2002)) is the most well-known algorithm of this type. SMOTE first finds a minority class sample at random and selects its k nearest minority class neighbors (typically $k = 5$). Then, a synthetic example is created as a convex combination of the minority class sample and a randomly selected neighbor. Batista et al. (2004) improves the quality of synthetic minority samples of SMOTE by removing the synthetic samples, whose nearest neighbor belongs to the majority class. He et al. (2008) consider a density distribution to decide the number of synthetic samples to be generated for a minority sample, whereas in SMOTE, there is a uniform weight for all minority points. Geometric Synthetic Minority Oversampling Technique (GSMOTE) is the state-of-art oversampling method that finds a geometric region around each minority class samples where synthetic data are generated inside this region (Douzas and Bacao (2019)). However, the oversampling method increases the number of training samples resulting an increase in the training time. It may also overfit on the test set as more emphasis is placed on specific minority samples.

Undersampling method is the third type of method to decrease the number of majority class to convert an imbalanced dataset to a more balanced dataset. Majority samples are removed randomly or based on their distance to minority samples (Kubat et al. (1997)). Wilson and Martinez (2000) remove the majority sample, whose class differs from the most of its three nearest neighbors. Smith et al. (2014) develop a method called Instance hardness threshold (IHT) that first train a classier and then delete the samples that have lower probabilities of belonging to the minority class. However, the undersampling method may result in loss of information as it discards potentially valuable data.

Recently, RO was used for classification problems (Bertsimas et al. (2011)), it was used more for classification problems. Pant et al. (2011) first utilized RO to classification problem to improve the performance of SVM. Bertsimas, Dunn, Pawlowski and Zhuo (2019) considered the uncertainties in features and

in labels for both LR and SVM. However, these papers only address the data uncertainties but not address the imbalance problem.

In this paper, we use RO to address LR and SVM on imbalanced datasets. We illustrate how to use RO to construct a balanced training set for both LR and SVM. We show that the proposed method provides a significant performance edge for the F1-metric (6% $\sim$ 8%), a small performance edge on the AUC-metric ($\sim$ 1%) over the state-of-the-art oversampling and undersampling methods on 20 real-world imbalanced datasets, while its running time is higher, but still within practical range (the proposed methods runs in minutes for $n \sim 60,000$)

The paper is structured as follows. In Section 2, we describe how to use RO on LR and SVM. In Section 3, we present computational results on 20 real-world datasets and compare it with other methods, including weighted LR/SVM, IHT and GSMOTE. We conclude in Section 4.

2 Methods

In this section, we describe how we apply RO to train LR and SVM on an imbalanced dataset. We focus on binary classification problem. For multi-classification problem, we can use one-versus-rest to change it to binray classifications.

Given data $(\mathbf{x}_i, y_i)$, $i \in [n]$, where $[n] = \{1, \ldots, n\}$, with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \{0, 1\}$, we normalize them by applying the transformation

$$\hat{x}_{ij} := \frac{x_{ij} - m_j}{\sigma_j}, \ i \in [n], \ j \in [p], \tag{1}$$

where m_j and σ_j are the mean and standard deviation in dimension j of the training samples.

Assume there are n_1 Class 1 minority samples and n_0 Class 0 majority samples. For imbalanced datasets, n_0 is much larger than n_1 and IR is defined as $\frac{n_0}{n_1}$. We use $\mathbf{x_i^0}$ ($i \in [n_0]$) and $\mathbf{x_j^1}$ ($j \in [n_1]$) to represent Class 0 and Class 1 samples, respectively.

2.1 Logistic Regression

LR is one of the most widely used binary classification methods which can be solved efficiently for large-scale datasets (Friedman et al. (2010)). In the LR problem, we find coefficients $\boldsymbol{\beta} \in \mathbb{R}^p, \beta_0 \in \mathbb{R}$ by solving the following problem:

$$\max_{\boldsymbol{\beta}, \beta_0} \left[-\sum_{i=1}^{n_0} \log(1 + e^{(\boldsymbol{\beta}^T \mathbf{x_i^0} + \beta_0)}) - \sum_{j=1}^{n_1} \log(1 + e^{-(\boldsymbol{\beta}^T \mathbf{x_j^1} + \beta_0)}) \right]. \tag{2}$$

Similar to the regularization term in the popular ridge regression (Hoerl and Kennard (1970)), a L2 regularization term can be added to the original objective function

$$\max_{\boldsymbol{\beta},\beta_0}\left[-\sum_{i=1}^{n_0}\log(1+e^{(\boldsymbol{\beta}^T\mathbf{x_i^0}+\beta_0)})-\sum_{j=1}^{n_1}\log(1+e^{-(\boldsymbol{\beta}^T\mathbf{x_j^1}+\beta_0)})\right]-\lambda||\boldsymbol{\beta}||_2^2. \quad (3)$$

Typically, we randomly assign data to the training and validation sets, and we select the parameter λ by using the performance in the validation set.

Motivated by the work of Bertsimas and Paskov (2019) in the context of stable linear regression, we propose to optimally assign data to the training and validation set in order to address imbalanced classification problem as follows. We introduce binary variables $u_i, i \in [n_0]$ and $v_j, j \in [n_1]$ to decide whether we use data x_i^0 and x_j^1 in the training set. In order to ensure we use a balanced training set, we add constraints

$$\sum_{j=1}^{n_1} v_j = \lfloor tn_1 \rfloor, \quad \sum_{i=1}^{n_0} u_i = k\lfloor tn_1 \rfloor,$$

where k is a hyper-parameter that control the imbalance between Class 0 and Class 1, $t \in [0,1]$ is the portion of the cardinality of training set relative to the total number of points. We then solve the following robust optimization problem

$$\max_{\boldsymbol{\beta},\beta_0}\min_{\mathbf{u}\in\mathcal{U},\mathbf{v}\in\mathcal{V}}\left[-\sum_{i=1}^{n_0}u_i\log(1+e^{(\boldsymbol{\beta}^T\mathbf{x_i^0}+\beta_0)})-\sum_{j=1}^{n_1}v_j\log(1+e^{-(\boldsymbol{\beta}^T\mathbf{x_j^1}+\beta_0)})\right]-\lambda||\boldsymbol{\beta}||_2^2 \quad (4)$$

with

$$\mathcal{U}=\left\{\mathbf{u}:\sum_{i=1}^{n_0}u_i=k\lfloor tn_1\rfloor, u_i\in\{0,1\}\right\}, \mathcal{V}=\left\{\mathbf{v}:\sum_{j=1}^{n_1}v_j=\lfloor tn_1\rfloor, v_j\in\{0,1\}\right\}.$$

As the inner minimization problem is linear in $\mathbf{u}$ and $\mathbf{v}$, it is equivalent to optimizing over the convex hull of $\mathcal{U}$ and $\mathcal{V}$

$$\text{conv}(\mathcal{U})=\{\mathbf{u}:\textstyle\sum_{i=1}^{n_0}u_i=k\lfloor tn_1\rfloor, 0\le u_i\le 1\}, \text{conv}(\mathcal{V})=\{\mathbf{v}:\textstyle\sum_{i=1}^{n_1}v_j=\lfloor tn_1\rfloor, 0\le v_i\le 1\}.$$

So Problem (4) is equivalent to

$$\max_{\boldsymbol{\beta},\beta_0}\min_{\mathbf{u}\in\text{conv}(\mathcal{U}),\mathbf{v}\in\text{conv}(\mathcal{V})}\left[-\sum_{i=1}^{n_0}u_i\log(1+e^{(\boldsymbol{\beta}^T\mathbf{x_i^0}+\beta_0)})-\sum_{j=1}^{n_1}v_j\log(1+e^{-(\boldsymbol{\beta}^T\mathbf{x_j^1}+\beta_0)})\right]-\lambda||\boldsymbol{\beta}||_2^2 \quad (5)$$

Note that Problem (5) seeks to find the hardest balanced training set with IR $= k$. Then, following RO techniques, we calculate the robust counterpart

of the inner problem

$$\begin{aligned}
&\min_{u,v} \left[-\sum_{i=1}^{n_0} u_i \log(1+e^{(\boldsymbol{\beta}^T \mathbf{x_i^0}+\beta_0)}) - \sum_{j=1}^{n_1} v_j \log(1+e^{-(\boldsymbol{\beta}^T \mathbf{x_j^1}+\beta_0)}) \right] \qquad (6)\\
\text{s.t.} \quad &\sum_{i=1}^{n_0} u_i = k\lfloor tn_1 \rfloor,\\
&\sum_{j=1}^{n_1} v_j = \lfloor tn_1 \rfloor,\\
&0 \le u_i, v_j \le 1, \quad i \in [n_0], \quad j \in [n_1],
\end{aligned}$$

by introducing the dual variables θ_0, θ_1 for the first constraint and the second constraint respectively, and dual variables p_i, q_i for the third set of constraints to get

$$\begin{aligned}
\max_{\theta_0,\theta_1,p,q} \quad & ktn_1\theta_0 + tn_1\theta_1 + \sum_{i=1}^{n_0} p_i + \sum_{j=1}^{n_1} q_j \qquad (7)\\
\text{s.t.} \quad & \theta_0 + p_i \le -\log(1+e^{(\boldsymbol{\beta}^T \mathbf{x_i^0}+\beta_0)}), \quad i \in [n_0],\\
& \theta_1 + q_j \le -\log(1+e^{-(\boldsymbol{\beta}^T \mathbf{x_j^1}+\beta_0)}), \quad j \in [n_1],\\
& p_i, q_j \le 0, \quad i \in [n_0], \quad j \in [n_1].
\end{aligned}$$

Substituting Problem (7) back into Problem (5), we obtain that Problem (4) is equivalent to:

$$\begin{aligned}
\max_{\boldsymbol{\beta},\beta_0,\theta_0,\theta_1,p,q} \quad & ktn_1\theta_0 + tn_1\theta_1 + \sum_{i=1}^{n_0} p_i + \sum_{j=1}^{n_1} q_j - \lambda||\boldsymbol{\beta}||_2^2 \qquad (8)\\
\text{s.t.} \quad & \theta_0 + p_i \le -\log(1+e^{(\boldsymbol{\beta}^T \mathbf{x_i^0}+\beta_0)}), \quad i \in [n_0],\\
& \theta_1 + q_j \le -\log(1+e^{-(\boldsymbol{\beta}^T \mathbf{x_j^1}+\beta_0)}), \quad j \in [n_1],\\
& p_i, q_j \le 0, \quad i \in [n_0], \quad j \in [n_1].
\end{aligned}$$

Because this maximization problem has a twice continuously differentiable concave objective function subject to convex constraints, we can solve it with interior point methods (Boyd and Vandenberghe (2004)).

In some extreme cases, the optimal solution of Problem (8) will be $\boldsymbol{\beta} = \mathbf{0}$ which is undesirable. In order to control proximity of the optimal solution to Problem (8) to the solution of Problem (2), we add constraints $|\beta_k - \hat{\beta}_k| \le \delta$

($k \in [p]$) to obtain:

$$\begin{aligned}
\max_{\boldsymbol{\beta},\beta_0,\theta_0,\theta_1,p,q} \quad & ktn_1\theta_0 + tn_1\theta_1 + \sum_{i=1}^{n_0} p_i + \sum_{j=1}^{n_1} q_j - \lambda||\boldsymbol{\beta}||_2^2 \\
\text{s.t.} \quad & \theta_0 + p_i \leq -\log(1 + e^{(\boldsymbol{\beta}^T \mathbf{x_i^0} + \beta_0)}), \quad i \in [n_0], \\
& \theta_1 + q_j \leq -\log(1 + e^{-(\boldsymbol{\beta}^T \mathbf{x_j^1} + \beta_0)}), \quad j \in [n_1], \\
& \beta_k - \hat{\beta}_k \leq \delta, \quad \beta_0 - \hat{\beta}_0 \leq \delta, \quad k \in [p], \\
& \hat{\beta}_k - \beta_k \leq \delta, \quad \hat{\beta}_0 - \beta_0 \leq \delta, \quad k \in [p], \\
& p_i, q_j \leq 0, \quad i \in [n_0], \quad j \in [n_1],
\end{aligned} \tag{9}$$

where $\hat{\boldsymbol{\beta}}, \hat{\beta}_0$ is the solution of weighted LR and δ is a small number decided by validation.

2.2 Support Vector Machines

SVM is another widely used binary classification method proposed by Cortes and Vapnik (1995). We find coefficients $\mathbf{w} \in \mathbb{R}^p, b \in \mathbb{R}$ by solving the following problem:

$$\min_{\mathbf{w},b} \left[\sum_{i=1}^{n_0} \max\left\{1 + \mathbf{w}^T \mathbf{x_i^0} - b, 0\right\} + \sum_{j=1}^{n_1} C \max\left\{1 - \mathbf{w}^T \mathbf{x_j^1} + b, 0\right\} \right] + \lambda||\mathbf{w}||_2^2 \tag{10}$$

where C is weight of Class 1 samples.

Similar to LR, the robust problem we solve is:

$$\min_{\mathbf{w},b} \max_{\mathbf{u}\in\mathcal{U},\mathbf{v}\in\mathcal{V}} \left[\sum_{i=1}^{n_0} u_i \max\left\{1 + \mathbf{w}^T \mathbf{x_i^0} - b, 0\right\} + \sum_{j=1}^{n_1} C v_j \max\left\{1 - \mathbf{w}^T \mathbf{x_j^1} + b, 0\right\} \right] + \lambda||\mathbf{w}||_2^2 \tag{11}$$

with

$$\mathcal{U} = \left\{ \mathbf{u} : \sum_{i=1}^{n_0} u_i = k\lfloor tn_1 \rfloor, u_i \in \{0,1\} \right\}, \mathcal{V} = \left\{ \mathbf{v} : \sum_{j=1}^{n_1} v_j = \lfloor tn_1 \rfloor, v_j \in \{0,1\} \right\}.$$

Replacing set $\mathcal{U}$ and $\mathcal{V}$ with their convex hull, Problem (11) is equivalent to:

$$\begin{aligned}
\min_{\mathbf{w},b} \max_{\mathbf{u},\mathbf{v}} \quad & \left[\sum_{i=1}^{n_0} u_i \max\left\{1 + \mathbf{w}^T \mathbf{x_i^0} - b, 0\right\} + \sum_{j=1}^{n_1} C v_j \max\left\{1 - \mathbf{w}^T \mathbf{x_j^1} + b, 0\right\} \right] + \lambda||\mathbf{w}||_2^2 \\
\text{s.t.} \quad & \sum_{i=1}^{n_0} u_i = k\lfloor tn_1 \rfloor, \quad \sum_{j=1}^{n_1} v_j = \lfloor tn_1 \rfloor, \quad 0 \leq u_i, v_j \leq 1.
\end{aligned} \tag{12}$$

Following RO techniques, we calculate the robust counterpart of the inner problem

$$\max_{\mathbf{u},v} \left[\sum_{i=1}^{n_0} u_i \max\left\{1 + \mathbf{w}^T \mathbf{x_i^0} - b, 0\right\} + \sum_{j=1}^{n_1} C v_j \max\left\{1 - \mathbf{w}^T \mathbf{x_j^1} + b, 0\right\} \right] \tag{13}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{i=1}^{n_0} u_i = k\lfloor tn_1 \rfloor, \\ & \sum_{j=1}^{n_1} v_j = \lfloor tn_1 \rfloor, \\ & 0 \le u_i, v_j \le 1, \end{aligned}$$

by introducing the dual variables θ_0, θ_1 for the first constraint and the second constraint respectively, and dual variables p_i, q_i for the third set of constraints to get

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \sum_{i=1}^{n_0} p_i + \sum_{j=1}^{n_1} q_j + k\lfloor tn_1 \rfloor \theta_0 + \lfloor tn_1 \rfloor \theta_1 \\ \text{s.t.} \quad & p_i + \theta_0 \ge 1 + \mathbf{w}^T \mathbf{x_i^0} - b, \quad i \in [n_0], \\ & p_i + \theta_0 \ge 0, \quad i \in [n_0], \\ & q_j + \theta_1 \ge C(1 - \mathbf{w}^T \mathbf{x_j^1} + b), \quad j \in [n_1], \\ & q_j + \theta_1 \ge 0, \quad j \in [n_1], \\ & p_i, q_j \ge 0, \quad i \in [n_0], \quad j \in [n_1]. \end{aligned} \tag{14}$$

Substituting Problem (14) back into Problem (12), we obtain that Problem (11) is equivalent to

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \sum_{i=1}^{n_0} p_i + \sum_{j=1}^{n_1} q_j + k\lfloor tn_1 \rfloor \theta_0 + \lfloor tn_1 \rfloor \theta_1 + \lambda ||\mathbf{w}||_2^2 \\ \text{s.t.} \quad & p_i + \theta_0 \ge 1 + \mathbf{w}^T \mathbf{x_i^0} - b, \quad i \in [n_0], \\ & p_i + \theta_0 \ge 0, \quad i \in [n_0], \\ & q_j + \theta_1 \ge C(1 - \mathbf{w}^T \mathbf{x_j^1} + b), \quad j \in [n_1], \\ & q_j + \theta_1 \ge 0, \quad j \in [n_1], \\ & p_i, q_j \ge 0, \quad i \in [n_0], \quad j \in [n_1], \end{aligned} \tag{15}$$

which is a convex quadratic problem.

Similar to LR, in order to control proximity of the optimal solution to Problem (15) to Problem (10), we also add constraints $|w_k - \hat{w}_k| \le \delta$ $(k \in [p])$

to obtain:

$$\begin{aligned}
\min_{\mathbf{w},b} \quad & \sum_{i=1}^{n_0} p_i + \sum_{j=1}^{n_1} q_j + k\lfloor tn_1 \rfloor \theta_0 + \lfloor tn_1 \rfloor \theta_1 + \lambda ||\mathbf{w}||_2^2 \qquad (16)\\
\text{s.t.} \quad & p_i + \theta_0 \geq 1 + \mathbf{w}^T \mathbf{x_i^0} - b, \quad i \in [n_0],\\
& p_i + \theta_0 \geq 0, \quad i \in [n_0],\\
& q_j + \theta_1 \geq C(1 - \mathbf{w}^T \mathbf{x_j^1} + b), \quad j \in [n_1],\\
& q_j + \theta_1 \geq 0, \quad j \in [n_1],\\
& w_k - \hat{w}_k \leq \delta, \quad b - \hat{b} \leq \delta, \quad k \in [p],\\
& \hat{w}_k - w_k \leq \delta, \quad \hat{b} - b \leq \delta, \quad k \in [p],\\
& p_i, q_j \geq 0, \quad i \in [n_0], \quad j \in [n_1],
\end{aligned}$$

where $\hat{\mathbf{w}}, \hat{b}$ is the solution of weighted SVM and δ is a small number decided by validation.

2.3 Validation Process

Under our robust framework, the training set represents the hardest set among the given data. Correspondingly, the validation is the easiest. For this reason, we use both the training and the validation set for validation. The whole process for training robust LR/SVM is as follows.

Algorithm 1 Robust LR/SVM on Imbalanced Datasets

Input: The whole dataset, parameters $\{k_1, \ldots, k_i\}, \{\lambda_1, \ldots, \lambda_j\}, \{\delta_1, \ldots, \delta_l\}$ to tune over.
Output: $\boldsymbol{\beta}, \beta_0$ for LR or $\mathbf{w}, b$ for SVM.

1: We randomly partition the whole dataset into two parts: the training set (75%) and the test set (25%).
2: **for** $k = k_1, \ldots, k_i$ **do**
3: **for** $\lambda = \lambda_1, \ldots, \lambda_j$ **do**
4: Solve Problem (8) for LR or Problem (15) for SVM with parameters k, λ and $t = 0.75$.
5: If the solution is $\boldsymbol{\beta} = \mathbf{0}$ for LR or $\mathbf{w} = \mathbf{0}$ for SVM,
6: **for** $\delta = \delta_1, \ldots, \delta_l$ **do**
7: Solve Problem (9) for LR or Problem (16) for SVM with parameters k, λ, δ instead.
8: **end for**
9: **end for**
10: **end for**
11: Identify the solution k^*, λ^*, δ^* with largest F1-score on the whole imbalanced training set.
12: Solve Problem (8) for LR or Problem (15) for SVM with selected parameters $k = k^*, \lambda = \lambda^*$ and $t = 1$.
13: If the solution is $\boldsymbol{\beta} = \mathbf{0}$ for LR or $\mathbf{w} = \mathbf{0}$ for SVM, we solve Problem (9) for LR or Problem (16) for SVM with parameters $k = k^*, \lambda = \lambda^*, \delta = \delta^*$ and $t = 1$ instead.
14: Return $\boldsymbol{\beta}, \beta_0$ for LR or $\mathbf{w}, b$ for SVM.

3 Computation Experiments

In this section, we compare the proposed method with oversampling, undersampling and weighted methods on a variety of imbalanced real-world datasets across two metrics: F1-score and Area Under Curve (AUC). F1-score is a measure of binary classifier's accuracy (Bradley (1997)), which is defined as:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{17}$$

AUC is another measure of accuracy, defined as the area under the Receiver Operating Characteristic (ROC) curve which usually used to measure the ability of a binary classifier system as its prediction threshold changes (Bradley (1997)).

3.1 Experiment Setup

We select 20 classification datasets from the UCI Machine Learning Repository (Dua and Graff (2017)). The datasets are selected with different sizes and imbalanced ratios. The IR ranges from 3.7 to 33.7, and the number of observations ranges from 79 to 245,057.

For multi-classification problem, we consider the one-versus-rest problem of predicting the occurrence of the minority class in the dataset. In seven of the data shown in Table 1 (flags-0, dermatology-5, connectionist-bench-10, yeast-me-2, statlog-project-landsat-satellite-5, chess-king-rook-vs-king-14), we identify the minority class in the dataset name. For example, for dataset with name flag-0, we predict Class 0 in this dataset. For non-robust methods, we partition each dataset into three parts: the training (50%), validation (25%), and test set (25%). First, we train the model using the training set with different hyper-parameters. Then, we select the best hyper-parameters using the validation set F1-score. After that, we retrain the model using the selected hyper-parameters on the combined training and validation sets. In the end, we report the F1-score and AUC on the test set. For the robust method, we combine the training and validation set as the new training set and use the validation process described in Section 2.3 to find the final solution. In the end, we report the F1-score and AUC on the test set. All methods are tested on the same random splits, and the experiments are repeated five times for each dataset with different random splits. We report the average test set F1-score and AUC across all five splits. For LR and SVM, we use the implementation from Pedregosa et al. (2011). The robust method is implemented using the JuMP software package in JULIA (Lubin and Dunning (2015)) and the solver IPOPT (Wächter and Biegler (2006)). For oversampling method GSMOTE, we use the implementation from Douzas and Bacao (2019). For undersampling method IHT, we use the implementation from Lemaître et al. (2017).

For weighted LR and SVM, we tune the regularization parameter $\lambda \in \{0.1, 1\}$ and set the weight of minority class to be IR. For IHT, we tune the

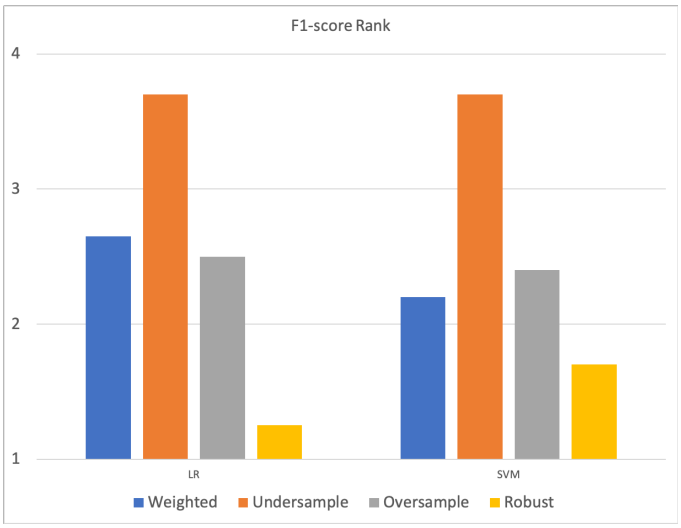

Fig. 1 F1-score rank for different methods.

regularization parameter $\lambda \in \{0.1, 1\}$. For GSMOTE, we tune the truncation factor from $\{-1.0, -0.5, 0, 0.25, 0.5, 0.75, 1\}$, the number of nearest neighbors $k \in \{3, 5\}$ and the deformation factor from $\{0.0, 0.2, 0.4, 0.6, 0.8, 1\}$ based on the suggestions from Douzas and Bacao (2019). For the robust method, we tune the ratio k from $\left\{1, \frac{1}{4}IR, \frac{1}{2}IR, \frac{3}{4}IR\right\}$, the regularization parameter $\lambda \in \{0.1, 1\}$ and $\delta \in \{0.05, 0.1\}$.

3.2 Comparisons

First, we compare the F1-score of the robust method with other methods. Tables 1 and 2 show the out-of-sample F1-score for LR and SVM, respectively. Rank 1 indicates the method that has the best performance on a dataset, and Rank 4 indicates the method that performs the worst. The column entitled Weighted represents the performance of weighted LR and SVM. The column entitled Undersample represents the performance of IHT, and the column entitled Oversample represents the performance of GSMOTE. The column entitled Robust represents the method proposed in this paper. For LR, the robust method increases out-of-sample F1-score by a 0.046 (8.0%) on average compared to weighted LR and a 0.045 (7.8%) on average compared to oversampling method GSMOTE. For SVM, the robust method increases out-of-sample F1-score by a 0.033 (5.7%) on average compared to weighted SVM and a 0.039 (6.8%) on average compared to oversampling method GSMOTE. Figure 1 shows the F1-score rank for different methods which illustrates that the robust method is the strongest method in terms of average rank, followed by GSMOTE and the weighted method.

UCI Dataset Name	n	p	IR	Weighted	Undersample	Oversample	Robust
hepatitis	79	20	5.1	0.655 (2)	0.466 (4)	0.593 (3)	**0.671** (1)
fertility	99	13	7.2	0.170 (4)	0.178 (3)	**0.247** (1)	0.19 (2)
flags-0	193	60	3.8	0.656 (3)	0.514 (4)	0.674 (2)	**0.709** (1)
image-segmentation-path	209	20	6	0.951 (3)	0.751 (4)	0.953 (2)	**0.982** (1)
dermatology-5	357	35	6.4	**1** (2)	0.543 (4)	**1** (2)	**1** (2)
libras-move	360	91	14	0.845 (2)	0.214 (4)	**0.849** (1)	0.651 (3)
thoracic-surgery	469	25	5.7	0.25 (3)	**0.266** (1)	0.24 (4)	0.257 (2)
climate-model-simulation-crashes	539	19	11	0.540 (2)	0.314 (4)	0.536 (3)	**0.764** (1)
connectionist-bench-10	989	11	10	0.194 (4)	0.212 (2)	0.195 (3)	**0.212** (1)
yeast-me-2	1484	9	28.1	0.209 (3)	0.083 (4)	0.218 (2)	**0.300** (1)
car-eval-34	1728	22	11.9	0.895 (2)	0.245 (4)	0.883 (3)	**0.915** (1)
ozone-level	2536	73	33.7	0.277 (3)	0.096 (4)	0.292 (2)	**0.348** (1)
sick-euthyroid	3163	43	9.8	0.640 (2)	0.223 (4)	0.639 (3)	**0.734** (1)
statlog-project-landsat-satellite-5	4434	37	8.4	0.384 (3)	0.287 (4)	0.385 (2)	**0.529** (1)
wine-quality	4898	12	25.8	0.189 (3)	0.092 (4)	0.191 (2)	**0.223** (1)
optical-digits	5620	65	9.1	0.777 (2)	0.343 (4)	0.776 (3)	**0.83** (1)
letter-img	20000	17	26.2	0.562 (3)	0.1 (4)	0.563 (2)	**0.793** (1)
chess-king-rook-vs-king-14	28055	35	5.2	0.509 (2)	0.37 (4)	0.509 (3)	**0.513** (1)
shuttle	58000	10	3.7	0.939 (2)	0.632 (4)	0.934 (3)	**0.948** (1)
skin-segmentation	245056	4	3.8	0.849 (3)	0.808 (4)	0.849 (2)	**0.859** (1)
Average F1score				0.575	0.337	0.576	0.621
Average rank				2.65	3.7	2.4	1.25

Table 1 F1-score and rank for different type of LR on each of the 20 datasets.

Next, we compare the AUC of all methods. Tables 3 and 4 show the out-of-sample AUC for LR and SVM, correspondingly. For LR, the robust method increases out-of-sample AUC by a 0.009 (1.0%) on average compared to weighted LR and a 0.005 (0.6%) on average compared to oversampling method GSMOTE. For SVM, the robust method increases out-of-sample AUC by a 0.01 (1.0%) on average compared to weighted SVM and a 0.008 (0.9%) on average compared to oversampling method GSMOTE. Figure 2 shows the AUC rank for different methods which illustrates the robust method is the strongest method in terms of average rank, followed by GSMOTE and the weighted method. The exact counts of wins, ties and losses for the robust method compared to other methods are shown in Table 6 and 7.

3.3 Computationally Tractability and Speed

The robust method does not change the nature of the optimization problem complexity. By adding robustness, LR changes from unconstrained convex optimization to constrained convex optimization. SVM remains quadratic optimization.

In order to better understand the scalability of the robust method, we also compare the total time of all methods for LR and SVM in selected UCI

UCI Dataset Name	n	p	IR	Weighted	Undersample	Oversample	Robust
hepatitis	79	20	5.1	0.641 (2)	0.505 (4)	0.621 (3)	**0.710** (1)
fertility	99	13	7.2	0.24 (3)	0.245 (2)	**0.247** (1)	0.147 (4)
flags-0	193	60	3.8	0.643 (3)	0.485 (4)	0.673 (2)	**0.750** (1)
image-segmentation-path	209	20	6	**0.982** (1)	0.725 (4)	0.94 (2)	0.935 (3)
dermatology-5	357	35	6.4	0.979 (3)	0.502 (4)	0.99 (2)	**1** (1)
libras-move	360	91	14	**0.907** (1)	0.222 (4)	0.854 (2)	0.638 (3)
thoracic-surgery	469	25	5.7	0.249 (2)	0.237 (3)	0.232 (4)	**0.268** (1)
climate-model-simulation-crashes	539	19	11	0.548 (2)	0.306 (4)	0.52 (3)	**0.696** (1)
connectionist-bench-10	989	11	10	0.197 (2)	**0.202** (1)	0.186 (3)	0.158 (4)
yeast-me-2	1484	9	28.1	0.213 (3)	0.078 (4)	0.22 (2)	**0.302** (1)
car-eval-34	1728	22	11.9	**0.908** (1)	0.255(4)	0.906 (2)	0.899 (3)
ozone-level	2536	73	33.7	0.277 (2)	0.102 (4)	0.274 (3)	**0.378** (1)
sick-euthyroid	3163	43	9.8	0.631 (3)	0.226 (4)	0.632 (2)	**0.748** (1)
statlog-project-landsat-satellite-5	4434	37	8.4	0.387 (2)	0.287 (4)	0.377 (3)	**0.456** (1)
wine-quality	4898	12	25.8	0.195 (3)	0.090 (4)	0.197 (2)	**0.230** (1)
optical-digits	5620	65	9.1	0.778 (2)	0.323 (4)	0.769 (3)	**0.833** (1)
letter-img	20000	17	26.2	0.559 (3)	0.099 (4)	0.56 (2)	**0.794** (1)
chess-king-rook-vs-king-14	28055	35	5.2	0.505 (3)	0.365 (4)	0.506 (2)	**0.509** (1)
shuttle	58000	10	3.7	0.926 (2)	0.640 (4)	0.925 (3)	**0.956** (1)
skin-segmentation	245056	4	3.8	**0.865** (1)	0.808 (4)	0.865 (2)	0.865 (3)
Average F1score				0.581	0.335	0.575	0.614
Average rank				2.2	3.7	2.4	1.7

Table 2 F1-score and rank for different type of SVM on each of the 20 datasets.

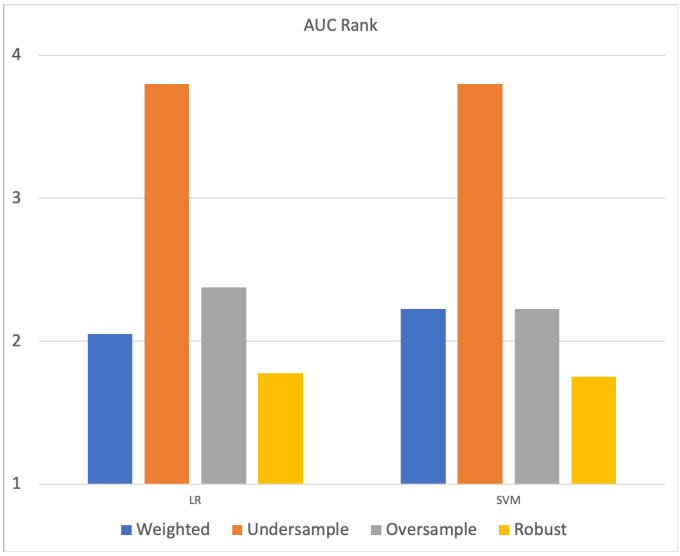

Fig. 2 AUC rank for different methods.

UCI Dataset Name	n	p	IR	Weighted	Undersample	Oversample	Robust
hepatitis	79	20	5.1	0.934 (3)	0.909 (4)	**0.957** (1)	0.943 (2)
fertility	99	13	7.2	0.433 (3)	0.382 (4)	**0.513** (1)	0.508 (2)
flags-0	193	60	3.8	**0.915** (1)	0.881 (4)	0.902 (3)	0.912 (2)
image-segmentation-path	209	20	6	**1** (2)	0.990 (4)	**1** (2)	**1** (2)
dermatology-5	357	35	6.4	**1** (2.5)	**1** (2.5)	**1** (2.5)	**1** (2.5)
libras-move	360	91	14	0.955 (2)	0.94 (4)	**0.957** (1)	0.948 (3)
thoracic-surgery	469	25	5.7	0.612 (3)	0.616 (2)	0.601 (4)	**0.619** (1)
climate-model-simulation-crashes	539	19	11	0.938 (2)	0.900 (4)	0.938 (3)	**0.975** (1)
connectionist-bench-10	989	11	10	0.612 (2)	0.559 (4)	0.608 (3)	**0.649** (1)
yeast-me-2	1484	9	28.1	0.837 (3)	0.822 (4)	0.842 (2)	**0.866** (1)
car-eval-34	1728	22	11.9	0.998 (2)	0.998 (3.5)	0.998 (3.5)	**0.999** (1)
ozone-level	2536	73	33.7	0.870 (3)	0.866 (4)	**0.887** (1)	0.883 (2)
sick-euthyroid	3163	43	9.8	0.947 (2)	0.919 (4)	0.947 (3)	**0.950** (1)
statlog-project-landsat-satellite-5	4434	37	8.4	**0.853** (1)	0.824 (4)	0.852 (2)	0.851 (3)
wine-quality	4898	12	25.8	**0.786** (1)	0.765 (4)	0.782 (3)	0.782 (2)
optical-digits	5620	65	9.1	0.981 (2)	0.977 (4)	0.981 (3)	**0.982** (1)
letter-img	20000	17	26.2	**0.989** (1)	0.987 (4)	0.989 (2)	0.989 (3)
chess-king-rook-vs-king-14	28055	35	5.2	0.83 (2)	0.808 (4)	0.83 (3)	**0.832** (1)
shuttle	58000	10	3.7	**0.992** (1)	0.983 (4)	0.992 (2)	0.991 (3)
skin-segmentation	245056	4	3.8	0.948 (2.5)	0.947 (4)	0.948 (2.5)	**0.948** (1)
Average AUC				0.872	0.854	0.876	0.881
Average rank				2.05	3.8	2.375	1.775

Table 3 AUC and rank for different type of LR on each of the 20 datasets.

datasets. All problems are solved using the MIT Engaging Cluster (Intel Xeon 2.1 GHz) with 1 CPU core and 16GB of memory. The results in 6 of 20 are shown in Table 5. They are selected based on different IR and sizes. For LR, the robust method slows down computation by one to two orders of magnitude, which is caused by solving a constrained convex optimization problem. For SVM, the robust method takes longer than the weighted and undersampling method but close to the oversampling method. The results here show that the robust method can be solved in minutes.

4 Conclusions

In this paper, we propose a robust optimization framework for solving the imbalanced classification problem for LR and SVM by balancing the majority and minority classes. We applied this algorithm to 20 imbalanced datasets collected from the UCI machine learning repository. The proposed algorithm ranks first on average compared to the state-of-the-art oversampling and undersampling method, improves the out-of-sample performance on the F1-metrics significantly and on the AUC-metrics slightly, while its running time is in minutes for problem $n \sim 60,000$.

UCI Dataset Name	n	p	IR	Weighted	Undersample	Oversample	Robust
hepatitis	79	20	5.1	0.936 (3)	0.886 (4)	0.941 (2)	**0.942** (1)
fertility	99	13	7.2	0.470 (4)	0.494 (2)	**0.497** (1)	0.478 (3)
flags-0	193	60	3.8	0.871 (3)	0.853 (4)	0.894 (2)	**0.911** (1)
image-segmentation-path	209	20	6	**1** (2)	0.996 (4)	**1** (2)	**1** (2)
dermatology-5	357	35	6.4	**1** (2)	0.998 (4)	**1** (2)	**1** (2)
libras-move	360	91	14	0.946 (2)	0.924 (4)	**0.947** (1)	0.940 (3)
thoracic-surgery	469	25	5.7	0.617 (2)	0.598 (4)	0.606 (3)	**0.674** (1)
climate-model-simulation-crashes	539	19	11	0.938 (3)	0.877 (4)	0.943 (2)	**0.962** (1)
connectionist-bench-10	989	11	10	0.615 (2)	0.585 (4)	0.603 (3)	**0.621** (1)
yeast-me-2	1484	9	28.1	0.841 (3)	0.801 (4)	0.843 (2)	**0.870** (1)
car-eval-34	1728	22	11.9	0.998 (2)	0.997 (4)	0.998 (3)	**0.998** (1)
ozone-level	2536	73	33.7	0.869 (4)	0.873 (2)	0.871 (3)	**0.891** (1)
sick-euthyroid	3163	43	9.8	0.95 (2)	0.917 (4)	0.949 (3)	**0.951** (1)
statlog-project-landsat-satellite-5	4434	37	8.4	0.852 (2)	0.830 (4)	0.85 (3)	**0.86** (1)
wine-quality	4898	12	25.8	**0.788** (1)	0.773 (4)	0.78 (3)	0.785 (2)
optical-digits	5620	65	9.1	**0.981** (1)	0.975 (4)	0.981 (2)	0.98 (3)
letter-img	20000	17	26.2	**0.989** (1)	0.986 (4)	0.989 (2)	0.987 (3)
chess-king-rook-vs-king-14	28055	35	5.2	**0.831** (1)	0.805 (4)	0.83 (2)	0.829 (3)
shuttle	58000	10	3.7	0.991 (2)	0.983 (4)	**0.991** (1)	0.991 (3)
skin-segmentation	245056	4	3.8	0.947 (2.5)	0.947 (4)	0.947 (2.5)	**0.947** (1)
Average AUC				0.871	0.855	0.873	0.881
Average rank				2.225	3.8	2.225	1.75

Table 4 AUC score and rank for different type of SVM on each of the 20 datasets.

		UCI dataset (number of points; dimension; IR)					
		fertility	connectionist-bench-10	yeast-me-2	sick-euthyroid	letter-img	shuttle
Model	Method	(99;13;7)	(989;11;10)	(1484;9;28)	(3163;43;10)	(20000;17;26)	(58000;10;4)
LR	Weighted	0.02	0.02	0.02	0.06	0.14	0.48
	Undersample	0.06	0.08	0.10	0.32	0.6	2.10
	Oversample	1.24	5.03	8.03	22.69	111.77	391.22
	Robust	19.38	35.72	34.45	518.32	776.52	2008.10
SVM	Weighted	0	0	0.02	0.06	0.10	0.44
	Undersample	0.04	0.48	0.64	4.12	23.34	103.66
	Oversample	0.65	4.77	7.64	21.96	106.36	370.51
	Robust	0.54	9.08	5.70	40.5	360.01	906.52

Table 5 Solving time for selected UCI datasets in seconds.

References

Batista, G. E., Prati, R. C. and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data, *ACM SIGKDD Explorations Newsletter* **6**(1): 20–29.

Bertsimas, D., Brown, D. B. and Caramanis, C. (2011). Theory and applications of robust optimization, *SIAM Review* **53**(3): 464–501.

Bertsimas, D., Dunn, J., Pawlowski, C. and Zhuo, Y. D. (2019). Robust classification, *INFORMS Journal on Optimization* **1**(1): 2–34.

Model	Methods	Wins	Losses	Ties
LR	Robust vs Weighted	18	1	1
	Robust vs Undersample	19	1	0
	Robust vs Oversample	17	2	1
SVM	Robust vs Weighted	14	6	0
	Robust vs Undersample	18	2	0
	Robust vs Oversample	14	6	0

Table 6 Pairwise comparisons of the robust method with other methods on F1-score.

Model	Methods	Wins	Losses	Ties
LR	Robust vs Weighted	12	6	2
	Robust vs Undersample	19	0	1
	Robust vs Oversample	11	7	2
SVM	Robust vs Weighted	12	6	2
	Robust vs Undersample	19	1	0
	Robust vs Oversample	12	6	2

Table 7 Pairwise comparisons of the robust method with other methods on AUC.

Bertsimas, D., Dunn, J., Steele, D. W., Trikalinos, T. A. and Wang, Y. (2019). Comparison of machine learning optimal classification trees with the pediatric emergency care applied research network head trauma decision rules, *JAMA Pediatrics* **173**(7): 648–656.

Bertsimas, D. and Paskov, I. (2019). Stable regression: On the power of optimization over randomization in training regression problems. submitted to *Journal of Machine Learning Research*.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*, Cambridge University Press.

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms, *Pattern Recognition* **30**(7): 1145–1159.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research* **16**: 321–357.

Chawla, N. V., Lazarevic, A., Hall, L. O. and Bowyer, K. W. (2003). Smoteboost: Improving prediction of the minority class in boosting, *European Conference on Principles of Data Mining and Knowledge Discovery*, Springer, pp. 107–119.

Cortes, C. and Vapnik, V. (1995). Support-vector networks, *Machine Learning* **20**(3): 273–297.

Douzas, G. and Bacao, F. (2019). Geometric smote a geometrically enhanced drop-in replacement for smote, *Information Sciences* **501**: 118–135.

Dua, D. and Graff, C. (2017). Uci machine learning repository (2017), *URL http://archive. ics. uci. edu/ml* **37**.

Friedman, J., Hastie, T. and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent, *Journal of Statistical Software* **33**(1): 1.

He, H., Bai, Y., Garcia, E. A. and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning, *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress On Computational Intelligence)*, IEEE, pp. 1322–1328.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* **12**(1): 55–67.

Kubat, M., Matwin, S. et al. (1997). Addressing the curse of imbalanced training sets: one-sided selection, *ICML*, Vol. 97, Nashville, USA, pp. 179–186.

Lemaître, G., Nogueira, F. and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning, *Journal of Machine Learning Research* **18**(17): 1–5.
URL: *http://jmlr.org/papers/v18/16-365.html*

Lubin, M. and Dunning, I. (2015). Computing in operations research using julia, *INFORMS Journal on Computing* **27**(2): 238–248.

Pant, R., Trafalis, T. B. and Barker, K. (2011). Support vector machine classification of uncertain and imbalanced data using robust optimization, *Proceedings of the 15th WSEAS International Conference on Computers*, World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, pp. 369–374.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**: 2825–2830.

Provost, F. and Fawcett, T. (2001). Robust classification for imprecise environments, *Machine Learning* **42**(3): 203–231.

Smith, M. R., Martinez, T. and Giraud-Carrier, C. (2014). An instance level analysis of data complexity, *Machine Learning* **95**(2): 225–256.

Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical Programming* **106**(1): 25–57.

Wilson, D. R. and Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms, *Machine Learning* **38**(3): 257–286.

Yang, Q. and Wu, X. (2006). 10 challenging problems in data mining research, *International Journal of Information Technology & Decision Making* **5**(04): 597–604.