Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Weighted logistic regression for large-scale imbalanced and rare events data

Maher Maalouf[a,*], Mohammad Siddiqi[b]

[a] Industrial & Systems Engineering, Khalifa University, P.O. Box 127788, Abu Dhabi, United Arab Emirates
[b] Aerospace & Mechanical Engineering, Khalifa University, P.O. Box 127788, Abu Dhabi, United Arab Emirates

ARTICLE INFO

Article history:
Received 22 September 2013
Received in revised form 14 January 2014
Accepted 14 January 2014
Available online 27 January 2014

Keywords:
Classification
Endogenous sampling
Logistic regression
Kernel methods
Truncated Newton

ABSTRACT

Latest developments in computing and technology, along with the availability of large amounts of raw data, have led to the development of many computational techniques and algorithms. Concerning binary data classification in particular, analysis of data containing rare events or disproportionate class distributions poses a great challenge to industry and to the machine learning community. Logistic Regression (LR) is a powerful classifier. The combination of LR and the truncated-regularized iteratively re-weighted least squares (TR-IRLS) algorithm, has provided a powerful classification method for large data sets. This study examines imbalanced data with binary response variables containing many more non-events (zeros) than events (ones). It has been established in the literature that these variables are difficult to predict and explain. This research combines rare events corrections to LR with truncated Newton methods. The proposed method, Rare Event Weighted Logistic Regression (RE-WLR), is capable of processing large imbalanced data sets at relatively the same processing speed as the TR-IRLS, however, with higher accuracy.

1. Introduction

In recent years, much attention in the machine learning community has been drawn to the problem of imbalanced or rare-events data. There are two main reasons for this. The first is that most of the traditional models and algorithms are based on the assumption that the classes in the data are balanced or evenly distributed. However, in many real-life applications the data is imbalanced, and when the imbalance is extreme, this problem is termed the *rare events* problem or the *imbalanced data* problem. Hence, the rare class presents several problems and challenges to existing classification algorithms [1,2].

The second reason for concern is the importance of rare events in real-life applications. By definition, rare events are occurrences that take place with a substantially lower frequency than commonly occurring events. Applications such as internet security [3], bankruptcy early warning systems and predictions [4,5] are gaining more importance in recent years. Other examples of rare events include fraudulent credit card transactions [6], word mispronunciation [7], tornadoes [8], telecommunication equipment failures [9], oil spills [10], international conflicts [11], state failure [12], landslides [13,14], train derailments [15] and rare events in a series of queues [16] among others.

King and Zeng [2] state that the problems associated with REs stem from two sources. First, when probabilistic statistical methods, such as Logistic Regression (LR), are used, they underestimate the probability of rare events, because they tend to be biased towards the majority class, which is the less important class. Second, commonly used data collection strategies are inefficient for rare events data. A dilemma exists between gathering more observations (instances) and including more informational, useful variables in the data set. When one of the classes represents a rare event, researchers tend to collect very large numbers of observations with very few explanatory variables in order to include as much data as possible for the rare class. This in turn could significantly increase the cost of data collection without boosting the underestimated probability of detecting the rare class or the rare event. King and Zeng [2] advocate under-sampling of the majority class when statistical methods such as LR are employed. They clearly demonstrated, however, that such designs are only consistent and efficient with the appropriate corrections.

Linear classification is an extremely important machine-learning and data-mining tool. Compared to other classification techniques, such as the kernel methods, which transform data into higher dimensional space, linear classifiers are implemented directly on data in their original space. The main advantage of linear classifiers is their efficient training and testing procedures, especially when

[*] Corresponding author. Tel.: +971 24018000.
E-mail addresses: maher.maalouf@kustar.ac.ae (M. Maalouf), mohammad.siddiqi@kustar.ac.ae (M. Siddiqi).

http://dx.doi.org/10.1016/j.knosys.2014.01.012
0950-7051/© 2014 Elsevier B.V. All rights reserved.

implemented on large and high-dimensional data sets [17]. Logistic regression [18,19], which is a linear classifier, has been proven to be a powerful classifier by providing probabilities and by extending to multi-class classification problems [20,21]. The advantages of using LR are that it has been extensively studied [22], and recently it has been improved through the use of truncated Newton's methods [23–27]. Furthermore, LR does not make assumptions about the distribution of the independent variables and it includes the probabilities of occurrences as a natural extension. Moreover, LR requires solving only unconstrained optimization problems. Hence, with the right algorithms, the computation time can be much less than that of other methods, such as Support Vector Machines (SVM) [28], which require solving a constrained quadratic optimization problem. Komarek [29] were the first to implement the truncated-regularized iteratively re-weighted least squares (TR-IRLS) on LR to classify large data sets, and they demonstrated that it can outperform the Support Vector Machines (SVM) algorithm. Later on, trust region Newton method [24], which is a type of truncated Newton, and truncated Newton interior-point methods [30] were applied for large scale LR problems.

The objective of this study is to provide a basis for solving problems with data that are at once large and imbalanced or rare-event data. This paper is an extension of the work proposed by Maalouf and Saleh [31], which introduces the implementation of LR rare-event corrections to the TR-IRLS algorithm. The algorithm proposed is termed Rare Event-Weighted Logistic Regression (RE-WLR), and is based on the RE-WKLR algorithm, developed by Maalouf and Trafalis [32]. The RE-WKLR is appropriate for small-to-medium size data sets in terms of both computational speed and accuracy. The ultimate objective is to gain significantly more accuracy in predictive REs with diminished bias and variance. Weighting, regularization, approximate numerical methods, bias correction, and efficient implementation are critical to enabling RE-WLR to be an effective and powerful method for predicting rare events in large data sets. Our analysis involves the standard multivariate cases in *finite* dimensional spaces.

In Section 2 we derive the LR model for the rare events and imbalanced data problems. Section 3 describes the Rare-Event Weighted Logistic Regression (RE-WLR) algorithm. Numerical results are presented in Section 4, and Section 5 addresses the conclusions and future work.

2. Logistic regression and sampling on the dependent variable

Let $\mathbf{X} \in \mathbb{R}^{N\times d}$ be a data matrix where N is the number of instances (examples) and d is the number of features (parameters or attributes), and $\mathbf{y}$ be a binary outcomes vector. For every instance $\mathbf{x}_i \in \mathbb{R}^d$ (a row vector in $\mathbf{X}$), where $i = 1 \dots N$, the outcome is either $y_i = 1$ or $y_i = 0$. Let the instances with outcomes of $y_i = 1$ belong to the positive class, and the instances with outcomes $y_i = 0$ belong to the negative class. The goal is to classify the instance $\mathbf{x}_i$ as positive or negative. An instance can be treated as a Bernoulli trial with an expected value $E(y_i)$ or probability p_i. The logistic function commonly used to model each instance $\mathbf{x}_i$ with its expected outcome is given by the following formula [22]:

$$E[y_i|\mathbf{x}_i,\beta] = p_i = \frac{e^{\mathbf{x}_i\beta}}{1+e^{\mathbf{x}_i\beta}}, \tag{1}$$

where β is the vector of parameters with the assumption that $x_{i0} = 1$ so that the intercept β_0 is a constant term. From then on, the assumption is that the intercept is included in the vector β.

The logistic (logit) transformation is the logarithm of the odds of the positive response and is defined as

$$\eta_i = \ln\left(\frac{p_i}{1-p_i}\right) = \mathbf{x}_i\beta. \tag{2}$$

In matrix form, the logit function is expressed as

$$\eta = \mathbf{X}\beta. \tag{3}$$

Now, with the assumption that the observations are independent, the likelihood function is

$$\mathbb{L}(\beta) = \prod_{i=1}^{\ell}(p_i)^{y_i}(1-p_i)^{1-y_i} = \prod_{i=1}^{\ell}\left(\frac{e^{\mathbf{x}_i\beta}}{1+e^{\mathbf{x}_i\beta}}\right)^{y_i}\left(\frac{1}{1+e^{\mathbf{x}_i\beta}}\right)^{1-y_i}. \tag{4}$$

The regularized log likelihood [22] is defined as

$$\log\mathbb{L}(\beta) = \sum_{i=1}^{\ell}\left(y_i\log\left(\frac{e^{\mathbf{x}_i\beta}}{1+e^{\mathbf{x}_i\beta}}\right) + (1-y_i)\log\left(\frac{1}{1+e^{\mathbf{x}_i\beta}}\right)\right) - \frac{\lambda}{2}\|\beta\|^2, \tag{5}$$

$$= -\sum\nolimits_{i=1}^{\ell}\log\left(e^{-y_i\mathbf{x}_i\beta}(1+e^{\mathbf{x}_i\beta})\right) - \frac{\lambda}{2}\|\beta\|^2, \tag{6}$$

where the regularization (penalty) term $\frac{\lambda}{2}\|\beta\|^2$ was added to obtain better generalization. Since the log likelihood function is strictly concave, the objective is then to find the Maximum Likelihood Estimate (MLE), $\hat{\beta}$, which maximizes the log likelihood. For binary outputs, the loss function or the deviance $\mathbb{DEV}$ is the negative log likelihood and is given by the formula [29,22]

$$\mathbb{DEV}(\hat{\beta}) = -2\ln\mathbb{L}(\beta). \tag{7}$$

Minimizing the deviance $\mathbb{DEV}(\hat{\beta})$ given in (7) is equivalent to maximizing the log-likelihood given in (2) [22]. The deviance function (above) is nonlinear in β. Minimizing it requires numerical methods in order to find the Maximum Likelihood Estimate (MLE) of β, which is $\hat{\beta}$. Recent studies have shown that the CG method provides better results to estimate β than any other numerical method [33,34].

When one of the $\mathbf{y}$ classes is rare in the population, then random selection within values of $\mathbf{y}$ would save significant resources in data collection [2,35]. Several advantages are associated with the selection on the response variable. First, in conducting surveys, cost reduction and time saving can be achieved by using stratified samples instead of collecting random samples, especially when the event of interest is rare in the population. Second, greater computational efficiency can be reached, because there is no need to analyze massive data sets. Finally, the explanatory power of the Logistic model can be enriched by making the proportions of events and non-events more balanced [2]. However, since the objective is to derive inferences about the population from the sample, the estimates obtained by the common likelihood using pure endogenous sampling are inconsistent. To see why this is so, under pure endogenous sampling, the conditioning is on $\mathbf{X}$ rather than $\mathbf{y}$ [36,37], and the joint distribution of $\mathbf{y}$ and $\mathbf{X}$ in the sample is

$$f_s(\mathbf{y},\mathbf{X}|\beta) = P_s(\mathbf{X}|\mathbf{y},\beta)P_s(\mathbf{y}), \tag{8}$$

where β is the unknown parameter vector to be estimated. Yet, since $\mathbf{X}$ is a matrix of exogenous variables, then the conditional probability of $\mathbf{X}$ in the sample is equal to that in the population, or $P_s(\mathbf{X}|\mathbf{y},\beta) = P(\mathbf{X}|\mathbf{y},\beta)$. However, the conditional probability in the population is

$$P(\mathbf{X}|\mathbf{y},\beta) = \frac{f(\mathbf{y},\mathbf{X}|\beta)}{P(\mathbf{yF})}, \tag{9}$$

but

$$f(\mathbf{y},\mathbf{X}|\beta) = P(\mathbf{y}|\mathbf{X},\beta)P(\mathbf{X}), \tag{10}$$

and hence, substituting and rearranging yields

$$f_s(\mathbf{y},\mathbf{X}|\beta) = \frac{P_s(\mathbf{y})}{P(\mathbf{y})}P(\mathbf{y}|\mathbf{X},\beta)P(\mathbf{X}), \tag{11}$$

$$= \frac{H}{Q}P(\mathbf{y}|\mathbf{X},\beta)P(\mathbf{X}), \tag{12}$$

where $\frac{H}{Q} = \frac{P_s(\mathbf{y})}{P(\mathbf{y})}$, with H representing the proportions in the sample and Q the proportions in the population. The likelihood is then

$$\mathbb{L}_{Endogenous} = \prod_{i=1}^{n} \frac{H_i}{Q_i} P(y_i|\mathbf{x}_i, \beta) P(\mathbf{x}_i). \tag{13}$$

Therefore, when dealing with REs and imbalanced data, the likelihood in (13) needs to be maximized [36,38–41]. Several consistent estimators of this type of likelihood have been proposed in the literature. Amemiya [38] and Ben Akiva and Lerman [42] provide an excellent survey of these methods.

Manski and Lerman [40] proposed the *Weighted Exogenous Sampling Maximum Likelihood* (WESML), and proved that WESML yields a consistent and asymptotically normal estimator so long as knowledge of the population probability is available. More recently, Ramalho and Ramalho [43] extended the work of Manski and Lerman [40] to cases where such knowledge may not be available. Knowledge of population probability or proportions, however, can be acquired from previous surveys or existing databases. The log-likelihood for LR can then be rewritten as

$$\ln \mathbb{L}(\beta|\mathbf{y}, \mathbf{X}) = \sum\nolimits_{i=1}^{n} \frac{Q_i}{H_i} \ln P(y_i|\mathbf{x}_i, \beta), \tag{14}$$

$$= \sum\nolimits_{i=1}^{n} \frac{Q_i}{H_i} \ln\left(\frac{e^{y_i \mathbf{x}_i \beta}}{1 + e^{\mathbf{x}_i \beta}}\right), \tag{15}$$

$$= \sum\nolimits_{i=1}^{n} w_i \ln\left(\frac{e^{y_i \mathbf{x}_i \beta}}{1 + e^{\mathbf{x}_i \beta}}\right),, \tag{16}$$

where $\frac{Q_i}{H_i} = \left(\frac{\tau}{\bar{y}}\right) y_i + \left(\frac{1-\tau}{1-\bar{y}}\right)(1 - y_i)$, with $\bar{y}$ and τ representing the proportion of events in the sample and in the population, respectively. Thus, in order to obtain consistent estimators, the likelihood is multiplied by the inverse of the fractions. The intuition behind weighting is that if the proportion of events in the sample is more than that in the population, then the ratio $\left(\frac{Q}{H}\right)$ is less than one, and hence the events are given less weight, while the non-events would be given more weight if their proportion in the sample is less than that in the population. The above estimator, however, is not fully efficient, because the information matrix equality does not hold. This is demonstrated as

$$-E\left[\frac{Q}{H} \nabla_\beta^2 \ln P(\mathbf{y}|\mathbf{X}, \beta)\right] \neq E\left[\left(\frac{Q}{H} \nabla_\beta \ln P(\mathbf{y}|\mathbf{X}, \beta)\right)\left(\frac{Q}{H} \nabla_\beta \ln P(\mathbf{y}|\mathbf{X}, \beta)\right)^{\mathrm{T}}\right], \tag{17}$$

and for the LR model it is

$$-\left[\frac{1}{n}\sum_{i=1}^{n}\left(\frac{Q_i}{H_i}\right) p_i(1 - p_i)\mathbf{x}_i\mathbf{x}_j\right] \neq \left[\frac{1}{n}\sum_{i=1}^{n}\left(\frac{Q_i}{H_i}\right)^2 p_i(1 - p_i)\mathbf{x}_i\mathbf{x}_j\right]. \tag{18}$$

Let $\mathbf{A} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Q_i}{H_i}\right) p_i(1 - p_i)\mathbf{x}_i\mathbf{x}_j$, and $\mathbf{B} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Q_i}{H_i}\right)^2 p_i(1 - p_i)\mathbf{x}_i\mathbf{x}_j$, then the asymptotic variance matrix of the estimator β is given by the *sandwich estimate*, such that $\mathbf{V}(\beta) = \mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}$ [38–40].

King and Zeng [2] extended the small-sample bias corrections, as described by McCullagh and Nelder [44], to include the weighted likelihood (16), demonstrating that even with choice-based sampling, these corrections can make a difference when the population probability of the event of interest is low. According to McCullagh and Nelder [44], and later Cordeiro and McCullagh [45], the bias vector is given by

$$bias(\hat{\beta}) = (\mathbf{X}^{\mathrm{T}}\mathbf{V}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{V}\xi, \tag{19}$$

where $\xi_i = Q_{ii}(\hat{p}_i - \frac{1}{2})$, and Q_{ii} are the diagonal elements of $\mathbf{Q} = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{V}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}$, which is the approximate covariance matrix of the logistic link function η. The second-order bias-corrected estimator is then

$$\tilde{\beta} = \hat{\beta} - bias(\hat{\beta}). \tag{20}$$

As for the variance matrix $\mathbf{V}(\tilde{\beta})$ of $\tilde{\beta}$, it is estimated using

$$\mathbf{V}(\tilde{\beta}) = \left(\frac{n}{n+d}\right)^2 \mathbf{V}(\hat{\beta}). \tag{21}$$

Since $\left(\frac{n}{n+d}\right)^2 < 1$, then $\mathbf{V}(\tilde{\beta}) < \mathbf{V}(\hat{\beta})$, and hence both the variance and the bias are now reduced.

The main advantage then of the bias correction method proposed by McCullagh and Nelder [44] is that it reduces both the bias and the variance [2]. The disadvantage is that it is corrective and not preventive, since it is applied after the estimation is complete, and hence it does not protect against infinite parameter values that arise from perfect separation between the classes [46,47]. This bias correction method can only be applied if the estimator, $\hat{\beta}$, has finite values.

Now, as mentioned earlier, LR regularization is used in the form of the ridge penalty $\frac{\lambda}{2}\|\beta\|^2$. When regularization is introduced, none of the coefficients is set to zero [48], and the problem of infinite parameter values is avoided. In addition, the importance of the parameter λ lies in determining the bias-variance trade-off of an estimator [49,50]. When λ is very small, there is less bias but more variance. On the other hand, larger values of λ would lead to more bias but less variance [51]. Therefore, the inclusion of regularization in the LR model is very important to reduce any potential inefficiency. However, as regularization carries the risk of a non-negligible bias, even asymptotically [51], the need for bias correction becomes inevitable [32]. In sum, bias correction is needed to account for any bias resulting from regularization, small samples, and rare events.

3. RE-WLR algorithm

Applying now the formulation suggested by King and Zeng [2] on the weighted LR to the WLR model, the weighted likelihood can be rewritten as

$$\mathbb{L}_W(\beta) = \prod_{i=1}^{n} (p_i)^{w_1 y_i}(1 - p_i)^{w_0(1-y_i)}, \tag{22}$$

where $w_1 = \frac{\tau}{\bar{y}}$, and $w_0 = \frac{1-\tau}{1-\bar{y}}$. Now,

$$p_i = E(y_i) = \left(\frac{1}{1 + e^{-\eta_i}}\right)^{w_1} \equiv p_i^{w_1}, \tag{23}$$

and hence,

$$p_i' = w_1 p_i^{w_1}(1 - p_i), \tag{24}$$

and

$$p_i'' = w_1 p_i^{w_1}(1 - p_i)(w_1 - (1 + w_1)p_i). \tag{25}$$

Finally, the bias vector for WLR can now be rewritten as

$$\mathbf{B}(\hat{\beta}) = (\mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{D}\xi, \tag{26}$$

where the *i*th element of the vector ξ is now

$$\xi_i = 0.5 Q_{ii}(1 + w_1 p_i - w_1), \tag{27}$$

with Q_{ii} as the diagonal elements of $\mathbf{Q}$, which is now $\mathbf{Q} = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}$, and $\mathbf{D} = diag(v_i w_i)$ for $i = 1 \ldots n$. The bias-corrected estimator becomes

$$\tilde{\beta} = \hat{\beta} - \mathbf{B}(\hat{\beta}). \tag{28}$$

For WLR, the gradient and Hessian are obtained by differentiating the regularized weighted log-likelihood,

$$\ln \mathbb{L}_W(\beta) = \sum_{i=1}^{n} w_i \ln \frac{e^{y_i \mathbf{x}_i \beta}}{1 + e^{\mathbf{x}_i \beta}} - \frac{\lambda}{2}\|\beta\|^2, \tag{29}$$

with respect to β. In matrix form, the gradient is

$$\nabla_{\beta} \ln \mathbb{L}_W(\beta) = \mathbf{X}^{\mathrm{T}}\mathbf{W}(\mathbf{y} - \mathbf{p}) - \lambda\beta, \tag{30}$$

where $\mathbf{W} = diag(w_i)$ and $\mathbf{p}$ is the probability vector whose elements are given in (1). The Hessian with respect to $\boldsymbol{\beta}$ is then

$$\nabla^2_{\beta} \ln \mathbb{L}_W(\beta) = -\mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{X} - \lambda\mathbf{I}. \tag{31}$$

The Newton–Raphson update with respect to β on the $(c+1)$th iteration is

$$\hat{\beta}^{(c+1)} = (\mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{z}^{(c)}, \tag{32}$$

where $\mathbf{z}^{(c)} = \mathbf{X}\hat{\beta}^{(c)} + \mathbf{D}^{-1}(\mathbf{y} - \mathbf{p})$ is the adjusted dependent variable or the adjusted response.

The *weighted least squares* (WLS) subproblem is then

$$(\mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{X} + \lambda\mathbf{I})\hat{\beta}^{(c+1)} = \mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{z}^{(c)}, \tag{33}$$

which is a system of linear equations with a matrix $\mathbf{X}$, a vector of adjusted responses $\mathbf{z}$, and a weight matrix $\mathbf{D}$. Both the weights and the adjusted response vector are dependent on $\hat{\beta}^{(c)}$, which is the current estimate of the parameter vector. Specifying an initial estimate $\hat{\beta}^{(0)}$ for $\hat{\beta}$ can be solved iteratively, giving a sequence of estimates that converges to the MLE of $\hat{\beta}$. This iterative process can be done using the CG method.

Like the TR-IRLS algorithm, in order to avoid the long computations that the CG may suffer from, a limit can be placed on the number of CG iterations, thus creating an approximate or truncated Newton direction.

Algorithm 1. WLR MLE Using IRLS

```
Data: $\mathbf{X}, \mathbf{y}, \hat{\beta}^{(0)}, w_1, w_0$
Result: $\hat{\beta}, \mathbf{B}(\hat{\beta}), \tilde{\beta}, \tilde{p}_i$
begin
    $c = 0$
    while $|\frac{DEV^{(c)} - DEV^{(c+1)}}{DEV^{(c+1)}}| > \varepsilon_1$ and $c \leqslant$ Max IRLS Iterations do
        for $i \leftarrow 1$ to $n$ do
            $\hat{p}_i = \frac{1}{1+e^{-\mathbf{x}_i\hat{\beta}}}$                          /* Compute probabilities */
            $v_i = \hat{p}_i(1-\hat{p}_i)$                                      /* Compute variance */
            $w_i = w_1 y_i + w_0(1-y_i)$                                  /* Compute weights */
            $z_i = \mathbf{x}_i\hat{\beta}^{(c)} + \frac{(y_i-\hat{p}_i)}{\hat{p}_i(1-\hat{p}_i)}$             /* Compute adjusted response */

        $Q = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}$             /* Compute the covariance matrix */
        $Q_{ii} = diag(Q)$                                                  /* */
        for $k \leftarrow 1$ to $n$ do
            $\xi_k = \frac{1}{2}Q_{ii_k}((1+w_1)\hat{p}_k - w_1)$              /* Compute the bias response */

        $\mathbf{D} = diag(v_i w_i)$                                            /* */
        $(\mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{X} + \lambda\mathbf{I})\hat{\beta}^{(c+1)} = \mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{z}^{(c)}$    /* Compute $\hat{\beta}$ via Algorithm 2 (CG) */
        $(\mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{X} + \lambda\mathbf{I})\mathbf{B}(\hat{\beta})^{(c+1)} = \mathbf{X}^{\mathrm{T}}\mathbf{D}\xi^{(c)}$    /* Compute $\mathbf{B}(\hat{\beta})$ via Algorithm 3 (CG) */
        $c = c + 1$
    $\tilde{\beta} = \hat{\beta} - \mathbf{B}(\hat{\beta})$                                  /* Compute the unbiased $\beta$ */
    $\tilde{p}_i = \frac{1}{1+e^{-\mathbf{X}_i\tilde{\beta}}}$                            /* Compute the optimal probabilities */

```

Algorithm 2. Linear CG for computing $\hat{\beta}$. $\mathbf{A} = \mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{X} + \lambda\mathbf{I}$, $\mathbf{b} = \mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{z}$

```
Data: $\mathbf{A}, \mathbf{b}, \hat{\beta}^{(0)}$
Result: $\hat{\beta}$ such that $\mathbf{A}\hat{\beta} = \mathbf{b}$
begin
    $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\hat{\beta}^{(0)}$                                   /* Initialize the residual */
    $c = 0$
    while $||\mathbf{r}^{(c+1)}||^2 > \varepsilon_2$ and $c \leqslant$ Max CG Iterations do
        if $c = 0$ then
            $\zeta^{(c)} = 0$
        else
            $\zeta^{(c)} = \frac{\mathbf{r}^{\mathrm{T}(c+1)}\mathbf{r}^{(c+1)}}{\mathbf{r}^{\mathrm{T}(c+1)}\mathbf{r}^{(c)}}$                  /* Update $\mathbf{A}$-Conjugacy enforcer */
        $\mathbf{d}^{(c+1)} = \mathbf{r}^{(c+1)} + \zeta^{(c)}\mathbf{d}^{(c)}$                   /* Update the search direction */
        $s^{(c)} = \frac{\mathbf{r}^{\mathrm{T}(c)}\mathbf{r}^{(c)}}{\mathbf{d}^{\mathrm{T}(c)}\mathbf{A}\mathbf{d}^{c}}$                        /* Compute the optimal step length */
        $\hat{\beta}^{(c+1)} = \hat{\beta}^{(c)} + \zeta^{(c)}\mathbf{d}^{(c+1)}$                 /* Obtain an approximate solution */
        $\mathbf{r}^{(c+1)} = \mathbf{r}^{(c)} - s^{(c)}\mathbf{A}\mathbf{d}^{(c+1)}$                 /* Update the residual */
        $c = c + 1$
```

Algorithm 3. Linear CG for computing the bias. $\mathbf{A} = \mathbf{X}^{\mathrm{T}}\mathbf{D}\mathbf{X} + \lambda\mathbf{I}, \mathbf{b} = \mathbf{X}^{\mathrm{T}}\mathbf{D}\xi$

```
Data: $\mathbf{A}, \mathbf{b}, \mathbf{B}(\hat{\beta})^{(0)}$
Result: $\mathbf{B}(\hat{\beta})$ such that $\mathbf{A}\mathbf{B}(\hat{\beta}) = \mathbf{b}$
begin
    $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{B}(\hat{\beta})^{(0)}$                /* Initialize the residual */
    $c = 0$
    while $||\mathbf{r}^{(c+1)}||^2 > \varepsilon_3$ and $c \leqslant$ Max CG Iterations do
        if $c = 0$ then
            $\zeta^{(c)} = 0$
        else
            $\zeta^{(c)} = \frac{\mathbf{r}^{\mathrm{T}(c+1)}\mathbf{r}^{(c+1)}}{\mathbf{r}^{\mathrm{T}(c+1)}\mathbf{r}^{(c)}}$        /* Update A-Conjugacy enforcer */
        $\mathbf{d}^{(c+1)} = \mathbf{r}^{(c+1)} + \zeta^{(c)}\mathbf{d}^{(c)}$              /* Update the search direction */
        $s^{(c)} = \frac{\mathbf{r}^{\mathrm{T}(c)}\mathbf{r}^{(c)}}{\mathbf{d}^{\mathrm{T}(c)}\mathbf{A}\mathbf{d}^{(c)}}$                  /* Compute the optimal step length */
        $\mathbf{B}(\hat{\beta})^{(c+1)} = \mathbf{B}(\hat{\beta})^{(c)} + \zeta^{(c)}\mathbf{d}^{(c+1)}$    /* Obtain approximate solution */
        $\mathbf{r}^{(c+1)} = \mathbf{r}^{(c)} - s^{(c)}\mathbf{A}\mathbf{d}^{(c+1)}$              /* Update the residual */
        $c = c + 1$
```

Similar to TR-IRLS termination criteria [23], for Algorithm 1 the maximum number of iterations is set to 30, and for the relative difference of deviance threshold, ε_1, the value 0.01 is found to be sufficient to reach the desired accuracy and at the same time maintain good convergence speed. As for Algorithms 2 and 3, the maximum number of iterations for the CG is set to 200 iterations. In addition, the CG convergence thresholds, ε_2 and ε_3, are set to 0.005. Furthermore, no more than three non-improving iterations are allowed on the CG algorithm.

4. Computational results and discussion

The performance of the RE-WLR algorithm is examined using five benchmark binary class data sets and a real-life tornado data set (see Table 1). The ds1.10 and ds1.100 data sets and their details are available in [52]. The state failure (SF) data set with its details can be found in [53]. The SF training data consists of 5921 instances, while the testing set consists of 1269 instances. Performance of the algorithm is then compared to that of TR-IRLS, which is implemented just as described by Komarek and Moore [23]. The values of the parameter λ which give the best generalization, were chosen from a range of different values (generally user-defined) and were tuned using the bootstrap method [54]. Just like RE-WKLR in Maalouf and Trafalis [32], the bootstrap method is applied only to the testing sets. Each bootstrap re-sample has the same size as the original sample [54].

For this study the total number of bootstrap rounds (B) are set to 1000 rounds on all of the data sets. This number of rounds of is found adequate to generate enough variations. The bootstrap accuracy (A) had at most a half width of its 95% confidence interval equal to 0.25. The bootstrap sample size is chosen equal to 10,000 on all of the data sets.

Table 1
Data sets.

	Instances	Features	Class		Rarity (%)
			0	1	
ds1.10	26,733	10	25,929	804	3
ds1.100	26,733	100	25,929	804	3
Covertype	31,500	54	30,000	1500	5
Cod-RNA	22,660	8	22,000	660	3
Financial	17,108	93	16,521	578	3
SF	1269	10	1242	27	2
Tornado	13,790	83	13,016	774	3

The overall bootstrap accuracy (A^*) was calculated according to the following. A sequence of sample accuracies,

$$a_1^{(\mathbf{1})}, \ldots, a_r^{(\mathbf{1})}, \ldots, a_B^{(\mathbf{1})}, a_1^{(\mathbf{0})}, \ldots, a_r^{(\mathbf{0})}, \ldots, a_B^{(\mathbf{0})},$$

was collected during the bootstrap procedure, where for a given round (r), $a_r^{(\mathbf{1})} = \frac{TP}{TP+FN}$ for class one, and $a_r^{(\mathbf{0})} = \frac{TN}{TN+FP}$ for class zero, where (TP) corresponds to the number of correctly classified positive instances (*true positive*), (FN) corresponds to the number of positive instances classified as negative (*false negative*), (FP) corresponds to the number of negative instances classified as positive (*false positive*), and (TN) corresponds to the number of correctly classified negative instances (*true negative*). After the bootstrap procedure is completed, the average accuracy of each class is computed. Then, for a given bootstrap procedure, the accuracy is

$$A = min\{a_{avg}^{(\mathbf{1})}, a_{avg}^{(\mathbf{0})}\}. \tag{34}$$

The overall accuracy reached, with different parameters, is considered to be $A^* = max\{A\}$. The interval between the 2.5th and 97.5th percentiles of the bootstrap distribution of a statistic is the non-parametric 95% bootstrap confidence interval. All of the data sets except for SF are preprocessed using normalization of a mean of zero and standard deviation of one. All of the computations for RE-WLR and TR-IRLS are carried out using MATLAB version 2012a on a 4 GiB RAM computer.

4.1. Benchmark data sets

The benchmark data sets are the ds1.10, ds1.100, Covertype, Cod-RNA, Financial and State Failure. The ds1.10 and ds1.100 data sets [52] are a compressed life sciences data sets. Each row of the original, expanded data set represented a chemistry or biology experiment, and the output represented the reactivity of the compound observed in the experiment. The number of principal components in these data is set to 10 for ds1.10 and to 100 in the case of ds1.100. The Forest CoverType data set [55] is used for prediction of forest cover type from cartographic variables only (no remotely sensed data). The data corresponds to four wilderness areas located in the Roosevelt National Forest of northern Colorado. These areas represent forests with minimal human-caused disturbances, so that existing forest cover types are more a result of ecological processes rather than forest management practices. The Cod-RNA data set [55] is used for detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change. The Financial data set was introduced in Pacheco et al. [56] and it describes financial ratios of firms to predict whether a firm is credit worthy or not. The state failure (SF) data set [53] describes the complete or partial collapse of a government based on a number of variables.

The aforementioned data sets are divided into training and testing sets. Two sampling schemes on the training data sets are applied. In the first, the training data sets were equally divided into 400 instances in each class for all data except for 15 instances in the SF data set (due to its size), chosen randomly from the training set, but the same instances are applied to all methods. In the second scheme, the number of events remained 400 ones in all data sets and 15 for SF, but the number of non-events is three times the number of ones (1200 instances but 45 instances in SF).

Tables 2 and 3 summarize the computation results for both methods with the balanced and imbalanced training sets, including their optimal λ and accuracy, respectively. With balanced training sets, Table 3 shows that the RE-WLR method scores better than TR-IRLS except for the ds1.10 data set where both algorithms achieve equal accuracy. With imbalanced training sets, TR-IRLS performs better only on the ds1.10 data sets while RE-WLR performed better on ds1.100, Covertype, and Cod-RNA. Both algorithms perform equally on both the Financial and SF data sets. When RE-WLR performs better, the difference in accuracy is significantly large (this was verified using paired t-test with significance level of $\alpha = 0.05$). In addition, the time difference between the two methods is not particularly large, as Table 4 indicates.

Table 2
Optimal parameter values for the balanced and imbalanced training data sets. λ is the regularization parameter for both RE-WLR and TR-IRLS.

	Balanced training set		Imbalanced training set	
	RE-WLR λ	TR-IRLS λ	RE-WLR λ	TR-IRLS λ
ds1.10	10^4	10^4	0.5	0.5
ds1.100	500	10^5	10	10^6
Covertype	100	10^5	10	10^5
Cod-RNA	0.1	0.01	100	0.001
Financial	10	10^4	10^5	10^5
SF	30	4×10^{-7}	10	60

Table 3
Bootstrap accuracy (%) using balanced and imbalanced training data sets. Bold accuracy values indicate the highest accuracy reached by the algorithms with statistical significance.

	Balanced training set		Imbalanced training set	
	RE-WLR	TR-IRLS	RE-WLR	TR-IRLS
ds1.10	55	55	55	**59**
ds1.100	**63**	60	**62**	59
Covertype	**48**	45	**52**	48
Cod-RNA	**73**	72	**71**	68
Financial	**67**	62	68	68
SF	**100**	48	100	100

Table 4
Comparison of bootstrap time in seconds using balanced and imbalanced training data sets.

	Balanced training set		Imbalanced training set	
	RE-WLR	TR-IRLS	RE-WLR	TR-IRLS
ds1.10	3.9	3.9	4.0	3.9
ds1.100	23.0	22.0	21.1	22.2
Covertype	72.0	70.8	80.3	72.5
Cod-RNA	4.2	3.2	5.6	3.2
Financial	19.3	17.3	19.4	17.5
SF	0.50	0.33	0.58	0.25

Table 5
Optimal parameter values for the balanced and imbalanced Tornado training data sets. C is the SVM regularization parameter, and λ is the regularization parameter for both RE-WLR and TR-IRLS.

	Balanced training set	Imbalanced training set
RE-WLR	$\lambda = 4$	$\lambda = 4$
SVM	$C = 1$	$C = 100$
TR-IRLS	$\lambda = 5$	$\lambda = 5$

Table 6
Bootstrap accuracy (%) using balanced and imbalanced Tornado training data sets. Bold accuracy values indicate the highest accuracy reached with statistical significance.

	Balanced training set	Imbalanced training set
RE-WLR	90$^\diamond$	91
SVM	90	91
TR-IRLS	83	90

$^\diamond$ Statistical significance using paired t-test with $\alpha = 0.017$ over TR-IRLS.

Table 7
Comparison of bootstrap time in seconds using balanced and imbalanced Tornado training data sets.

	Balanced training set	Imbalanced training set
RE-WLR	14.10	14.55
SVM	336.43	575.10
TR-IRLS	14.21	14.34

4.2. Tornado data set

The Tornado data set consists of 83 attributes, 24 of which are derived from the Mesocyclone Detection Algorithm (MDA) data, measuring radar-derived *velocity* parameters that describe aspects of the Mesocyclone in addition to the *month* attribute. The remaining attributes are from the Near Storm Environment (NSE) data [57], which describes the pre-storm environment on a broader scale than MDA data. The attributes of the NSE data consist of *wind speed*, *direction*, *wind shear*, *humidity lapse rate*, and the *predisposition* of the atmosphere to explosively lift air over specific heights. In addition, the original Tornado data set consists of a training set and testing set. The training set has 387 tornado observations and 1144 non-tornado observations. As with the benchmark data sets, the original Tornado data set is considered as the imbalanced training set, while an under-sampling scheme of 387 non-tornado instances (chosen randomly from the original training set) and 387 instances was carried out. The testing set consists of 387 tornado observations and 11,872 non-tornado observations, and hence the rarity is 3%.

In order to appreciate the speed and accuracy of the RE-WLR, the performance of the algorithm is compared to that of both Support Vector Machines (SVM) and TR-IRLS for the Tornado data set. In this analysis, the linear kernel,

$$K(x_i, x_j) = \langle x_i, x_j \rangle,$$

is used for the SVM method. The linear kernel is known to be the fastest kernel used by SVM [58]. For the SVM method, MATLAB LIBSVM toolbox (version 3.17) [59] is used In addition, statistical significance was established using a multiple comparison paired t-test [60] single tailed with an adjusted $\alpha = 0.017$.

As with the above analysis, Tables 5 and 6 summarize the results for these three methods with their optimal parameters and accuracy, respectively. Table 6 shows that RE-WLR performed just as well as SVM on the Tornado data set, on which it achieves equal accuracy. However, the computational speed of the RE-WLR algorithm, measured by CPU time as shown in Table 7, is tremendously faster than that of SVM, despite the fact that LIBSVM is programmed mainly in C++ while both the TR-IRLS and RE-WLR algorithms are coded purely in MATLAB. The time saving ranges between approximately 96% and 97% as indicated by Table 7.

5. Conclusions

We have presented the Rare Event Weighted Logistic Regression (RE-WLR) algorithm, which is based on the Rare Event Weighted Kernel Logistic Regression (RE-WLR) algorithm, and have shown that the RE-WLR algorithm is easy and robust when implemented on large imbalanced and rare event data, and it performed better than TR-IRLS. The algorithm combines several concepts from the fields of statistics, econometrics and machine learning. The RE-WLR algorithm utilizes bias correction and regularization. Future studies may implement the proposed algorithm on more data sets in order to ascertain its strength.

References

[1] G.M. Weiss, Mining with rarity: a unifying framework, SIGKDD Explor. Newslett. 6 (1) (2004) 7–19.

[2] G. King, L. Zeng, Logistic regression in rare events data, Polit. Anal. 9 (2001) 137–163.

[3] M. Hlosta, R. Stríž, J. Kupčík, J. Zendulka, T. Hruška, Constrained classification of large imbalanced data by logistic regression and genetic algorithm, Int. J. Mach. Learn. Comput. 2013 (3) (2013) 214–218.

[4] J. de Andrés, M. Landajo, P. Lorca, Bankruptcy prediction models based on multinorm analysis: an alternative to accounting ratios, Knowl.-Based Syst. 30 (2012) 67–77.

[5] C.-F. Tsai, K.-C. Cheng, Simple instance selection for bankruptcy prediction, Knowl.-Based Syst. 27 (0) (2012) 333–342.

[6] P.K. Chan, S.J. Stolfo, Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection, in: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1998, pp. 164–168.

[7] B. Busser, W. Daelemans, A. Bosch, Machine learning of word pronunciation: the case against abstraction, in: Proceedings of the Sixth European Conference on Speech Communication and Technology, Eurospeech99, ISCA, 1999, pp. 2123–2126.

[8] T.B. Trafalis, H. Ince, M.B. Richman, Tornado detection with support vector machines, in: P.M.A. Sloot, D. Abramson, A.V. Bogdanov, Y.E. Gorbachev, J.J. Dongarra, A.Y. Zomaya, (Eds.), Computational Science - ICCS 2003, in: Lecture Notes in Computer Science, vol. 2660, Springer Berlin Heidelberg, 2003, pp. 289–298.

[9] G.M. Weiss, H. Hirsh, Learning to predict extremely rare events, in: AAAI Workshop on Learning from Imbalanced Data Sets, AAAI Press, 2000, pp. 64–68.

[10] M. Kubat, R.C. Holte, S. Matwin, Machine learning for the detection of oil spills in satellite radar images, in: Machine Learning, Kluwer Academic Publishers, 1998, pp. 195–215.

[11] G. King, L. Zeng, Explaining rare events in international relations, Int. Organ. 55 (3) (2001) 693–715.

[12] G. King, L. Zeng, Improving forecast of state failure, World Polit. 53 (4) (2001) 623–658.

[13] M.V.D. Eeckhaut, T. Vanwalleghem, J. Poesen, G. Govers, G. Verstraeten, L. Vandekerckhove, Prediction of landslide susceptibility using rare events logistic regression: a case-study in the flemish ardennes (belgium), Geomorphology 76 (3–4) (2006) 392–410.

[14] S.B. Bai, J. Wang, F.Y. Zhang, A. Pozdnoukhov, M. Kanevski, Prediction of landslide susceptibility using logistic regression: a case study in bailongjiang river basin, China, Fourth Int. Conf. Fuzzy Syst. Knowl. Discovery 4 (2008) 647–651.

[15] J. Quigley, T. Bedford, L. Walls, Estimating rate of occurrence of rare events with empirical bayes: a railway application, Reliab. Eng. Syst. Saf. 92 (5) (2007) 619–627.

[16] P. Tsoucas, Rare events in series of queues, J. Appl. Probab. 29 (1992) 168–175.

[17] G.-X. Yuan, C.-H. Ho, C.-J. Lin, Recent advances of large-scale linear classification, Proc. IEEE 100 (9) (2012) 2584–2603.

[18] S. Canu, A.J. Smola, Kernel methods and the exponential family, Neurocomputing 69 (7-9) (2006) 714–720.

[19] T. Jaakkola, D. Haussler, Probabilistic kernel regression models, In: Proceedings of the 1999 Conference on AI and Statistics, Morgan Kaufmann, 1999.

[20] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer Verlag, 2001.

[21] P. Karsmakers, K. Pelckmans, J.A.K. Suykens, Multi-class kernel logistic regression: a fixed-size implementation, Int. Joint Conf. Neural Netw. (2007) 1756–1761.

[22] D.W. Hosmer, S. Lemeshow, Applied Logistic Regression, second ed., Wiley, 2000.

[23] P. Komarek, A. Moore, Making Logistic Regression a Core Data Mining Tool: A Practical Investigation of Accuracy, Speed, and Simplicity, Tech. Rep., Carnegie Mellon University, 2005.

[24] C.-J. Lin, R.C. Weng, S.S. Keerthi, Trust region newton methods for large-scale logistic regression, in: ICML '07: Proceedings of the 24th International Conference on Machine Learning, ACM, New York, NY, USA, 2007, pp. 561–568.

[25] M. Maalouf, T.B. Trafalis, Kernel logistic regression using truncated newton method, in: C.H. Dagli, D.L. Enke, K.M. Bryden, H. Ceylan, M. Gen (Eds.), Intelligent Engineering Systems Through Artificial Neural Networks, vol. 18, ASME Press, New York, NY, USA, 2008, pp. 455–462.

[26] M. Maalouf, T.B. Trafalis, I. Adrianto, Kernel logistic regression using truncated newton method, Comput. Manage. Sci. 8 (4) (2011) 415–428.

[27] M. Maalouf, Logistic regression in data analysis: an overview, Int. J. Data Anal. Tech. Strategies 3 (3) (2011) 281–299.

[28] V. Vapnik, The Nature of Statistical Learning, Springer, NY, 1995.

[29] P. Komarek, Logistic Regression for Data Mining and High-dimensional Classification, Ph.D. Thesis, Carnegie Mellon University, 2004.

[30] K. Koh, S. Kim, S. Boyd, An interior-point method for large-scale ℓ_1-regularized logistic regression, J. Mach. Learn. Res. 8 (2007) 1519–1555.

[31] M. Maalouf, H. Saleh, Weighted logistic regression for large-scale imbalanced and rare events data, in: A. Krishnamurthy, W. Chan (Eds.), Proceedings of the 2013 Industrial and Systems Engineering Research Conference (ISERC), Institute of Industrial Engineers (IIE) Annual Conference, San Juan, Puerto Rico, 2013.

[32] M. Maalouf, T.B. Trafalis, Robust weighted kernel logistic regression in imbalanced and rare events data, Comput. Stat. Data Anal. 55 (1) (2011) 168–183.

[33] T.P. Minka, A Comparison of Numerical Optimizers for Logistic Regression, Tech. Rep., Deptartment of Statistics, Carnegie Mellon University, 2003.

[34] R. Malouf, A comparison of algorithms for maximum entropy parameter estimation, in: Proceedings of Conference on Natural Language Learning, vol. 6, 2002.

[35] J.S. Cramer, Logit Models From Economics And Other Fields, Cambridge University Press, 2003.

[36] A.C. Cameron, P.K. Trivedi, Microeconometrics: Methods and Applications, Cambridge University Press, 2005.

[37] M. Milgate, J. Eatwell, P.K. Newman (Eds.), Econometrics, W.W. Norton & Company, 1990.

[38] T. Amemiya, Advanced Econometrics, Harvard University Press, 1985.

[39] Y. Xie, C.F. Manski, The logit model and response-based samples, Sociol. Methods Res. 17 (1989) 283–302.

[40] C.F. Manski, S.R. Lerman, The estimation of choice probabilities from choice based samples, Econometrica 45 (8) (1977) 1977–1988.

[41] G.W. Imbens, T. Lancaster, Efficient estimation and stratified sampling, J. Economet. 74 (1996) 289–318.

[42] M. Ben-Akiva, S. Lerman, Discrete Choice Analysis: Theory and Application to Travel Demand, The MIT Press, 1985.

[43] E.A. Ramalho, J.J.S. Ramalho, On the weighted maximum likelihood estimator for endogenous stratified samples when the population strata probabilities are unknown, Appl. Econ. Lett. 14 (2007) 171–174.

[44] P. McCullagh, J. Nelder, Generalized Linear Model, Chapman and Hall/CRC, 1989.

[45] G.M. Cordeiro, P. McCullagh, Bias correction in generalized linear models, J. Roy. Stat. Soc. 53 (3) (1991) 629–643.

[46] G. Heinze, M. Schemper, A solution to the problem of monotone likelihood in cox regression, Biometrics 57 (2001) 114–119.

[47] S. Wang, T. Wang, Precision of warm's weighted likelihood for a polytomous model in computerized adaptive testing, Appl. Psychol. Meas. 25 (4) (2001) 317–331.

[48] M.Y. Park, T. Hastie, Penalized logistic regression for detecting gene interactions, Biostatistics 9 (1) (2008) 30–50.

[49] G. Cowan, Statistical Data Analysis, Oxford University Press, 1998.

[50] O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, Springer, 2005.

[51] R. Berk, Statistical Learning from a Regression Perspective, first ed., Springer, 2008.

[52] P. Komarek. <http://komarix.org/ac/ds/>.

[53] G. King, L. Zeng, Replication Data for: Improving Forecasts of State Failure, 2001. <http://hdl.handle.net/1902.1/RPQIODIANRUNF:3:CEsbEgPxbxExfYuh2NWwWQ==IQSS> Dataverse Network [Distributor] V2 [Version].

[54] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall/CRC, 1994.

[55] A. Asuncion, D.J. Newman, UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, 2007. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

[56] J. Pacheco, S. Casado, L. Núñez, A variable selection method based on tabu search for logistic regression models, Eur. J. Oper. Res. 199 (2) (2009) 506–511.

[57] V. Lakshmanan, G. Stumpf, A. Witts, A neural network for detecting and diagnosing tornadic circulations using the mesocyclone detection and near storm environment algorithms, in: 21st International Conference on Information Processing Systems, American Meteorological Society, San Diego, CA, 2005 (CD-ROM, J52.2).

[58] T. Hendtlass, M. Ali (Eds.), Developments in Applied Artificial Intelligence, second ed., Springer, 2002.

[59] C.-C. Chang, C.-J. Lin, LIBSVM: A Library for Support Vector Machines, 2001. <http://www.csie.ntu.edu.tw/cjlin/libsvm>.

[60] D. Jensen, P.R. Cohen, Multiple comparison in induction algorithms, Mach. Learn. 38 (2000) 309–338.