Ensemble Approach for the Classification of Imbalanced Data

Conference Paper · November 2009

DOI: 10.1007/978-3-642-10439-8_30 · Source: OAI

Vladimir Nikulin
Vyatka State University
66 PUBLICATIONS **243** CITATIONS

SEE PROFILE

G. J. Mclachlan
The University of Queensland
416 PUBLICATIONS **34,936** CITATIONS

SEE PROFILE

Shu Kay Angus Ng
Griffith University
144 PUBLICATIONS **5,598** CITATIONS

SEE PROFILE

Ensemble Approach for the Classification of Imbalanced Data

Vladimir Nikulin[1], Geoffrey J. McLachlan[1], and Shu Kay Ng[2]

[1] Department of Mathematics, University of Queensland
v.nikulin@uq.edu.au, gjm@maths.uq.edu.au
[2] School of Medicine, Griffith University
s.ng@griffith.edu.au

Abstract. Ensembles are often capable of greater prediction accuracy than any of their individual members. As a consequence of the diversity between individual base-learners, an ensemble will not suffer from overfitting. On the other hand, in many cases we are dealing with imbalanced data and a classifier which was built using all data has tendency to ignore minority class. As a solution to the problem, we propose to consider a large number of relatively small and balanced subsets where representatives from the larger pattern are to be selected randomly. As an outcome, the system produces the matrix of linear regression coefficients whose rows represent random subsets and columns represent features. Based on the above matrix we make an assessment of how stable the influence of the particular features is. It is proposed to keep in the model only features with stable influence. The final model represents an average of the base-learners, which are not necessarily a linear regression. Test results against datasets of the PAKDD-2007 data-mining competition are presented.

Keywords: ensemble classifier, gradient-based optimisation, boosting, random forest, decision trees.

1 Introduction

Ensemble (including voting and averaged) classifiers are learning algorithms that construct a set of many individual classifiers (called base-learners) and combine them to classify test data points by sample average. It is now well-known that ensembles are often much more accurate than the base-learners that make them up [1], [2]. Tree ensemble called "random forest" was introduced in [3] and represents an example of successful classifier. Another example, bagging support vector machine (SVM) [4] is very important because direct application of the SVM to the whole data set may not be possible. In the case of SVM we are interested to deal with limited sample size which is equal to the dimension of the corresponding kernel matrix. The well known bagging technique [5] is relevant here. According to this technique each base-learner used in the ensemble is trained with data that are randomly selected from the training sample (without replacement).

A. Nicholson and X. Li (Eds.): AI 2009, LNAI 5866, pp. 291–300, 2009.

Our approach was motivated by [5], and represents a compromise between two major considerations. On the one hand, we would like to deal with balanced data. On the other hand, we are interested to exploit all available information. We consider a large number n of balanced subsets of available data where any single subset includes two parts 1) all 'positive' instances (minority) and 2) randomly selected 'negative' instances. The method of balanced random sets (RS) is general and may be used in conjunction with different base-learners.

In the experimental section we report test-results against real-world data of the PAKDD-2007 Data Mining Competition[1], which were provided by a consumer finance company with the aim of finding better solutions for a cross-selling problem. The data are strongly imbalanced with significantly smaller proportion of positive cases (1.49%), which have the following practical interpretation: a customer opened a home loan with the company within 12 months after opening the credit card [6].

Regularised linear regression (RLR) represents the most simple example of a decision function. Combined with quadratic loss function it has an essential advantage: using gradient-based search procedure we can optimise the value of the step size. Consequently, we will observe a rapid decline in the target function [7].

By definition, regression coefficients may be regarded as natural measurements of influence of the corresponding features. In our case we have n vectors of regression coefficients, and we can use them to investigate the stability of the particular coefficients.

Proper feature selection may reduce overfitting significantly [8]. We remove features with unstable coefficients, and recompute the classifiers. Note that stability of the coefficients may be measured using different methods. For example, we can apply the t-statistic given by the ratio of the mean to the standard deviation.

The proposed approach is flexible. We do not expect that a single algorithm will work optimally on all conceivable applications and, therefore, an opportunity of tuning and tailoring is a very essential.

Initial results obtained using RLR during PAKDD-2007 Data Mining Competition were reported in [9]. In this paper, using tree-based LogitBoost [10] as a base-learner we improved all results known to us.

This paper is organised as follows: Section 2 describes the method of random sets and mean-variance filtering. Section 3 discusses general principals of the AdaBoost and LogitBoost Algorithms. Section 4 explains the experimental procedure and the most important business insights. Finally, Section 5 concludes the paper.

2 Modelling Technique

Let $\mathbf{X} = (\mathbf{x}_t, y_t), t = 1, \ldots, m$, be a training sample of observations where $\mathbf{x}_t \in \mathbb{R}^\ell$ is a ℓ-dimensional vector of features, and y_t is a binary label: $y_t \in \{-1, 1\}$.

[1] http://lamda.nju.edu.cn/conf/pakdd07/dmc07/

Boldface letters denote vector-columns, whose components are labelled using a normal typeface.

In a practical situation the label y_t may be hidden, and the task is to estimate it using the vector of features. Let us consider the most simple linear decision function

$$u_t = u(\mathbf{x}_t) = \sum_{j=0}^{\ell} w_j \cdot x_{tj}, \tag{1}$$

where x_{t0} is a constant term.

We can define a decision rule as a function of decision function and threshold parameter

$$f_t = f(u_t, \mathit{\Delta}) = \begin{cases} 1 \text{ if } u_t \geq \mathit{\Delta}; \\ 0, \text{otherwise.} \end{cases}$$

We used AUC as an evaluation criterion where AUC is the area under the receiver operating curve (ROC). By definition, ROC is a graphical plot of True Positive Rates (TPR) against False Positive Rates (FPR).

According to the proposed method we consider large number of classifiers where any particular classifier is based on relatively balanced subset with all 'positive' and randomly selected (without replacement) 'negative' data. The final decision function (d.f.) has a form of logistic average of single decision functions.

Definition 1. *We call above subsets as random sets $RS(\alpha, \beta, n)$, where α is a number of positive cases, β is a number of negative cases, and n is the total number of random sets.*

This model includes two very important regulation parameters: 1) n and 2) $q = \frac{\alpha}{\beta} \leq 1$ - the proportion of positive cases where n must be sufficiently large, and q can not be too small.

We consider n subsets of $\mathbf{X}$ with α positive and $\beta = k \cdot \alpha$ negative data-instances, where $k \geq 1, q = \frac{1}{k}$. Using gradient-based optimization [11] we can compute the matrix of linear regression coefficients:

$$W = \{w_{ij}, i = 1, \ldots, n, j = 0, \ldots, \ell\}.$$

The mean-variance filtering (MVF) technique was introduced in [11], and may be efficient in order to reduce overfitting. Using the following ratios, we can measure the consistency of contributions of the particular features by

$$r_j = \frac{|\mu_j|}{s_j}, j = 1, \ldots, \ell, \tag{2}$$

where μ_j and s_j are the mean and standard deviation corresponding to the j-column of the matrix W.

A low value of r_j indicates that the influence of the j-feature is not stable. We conducted feature selection according to the condition:

$$r_j \geq \gamma > 0.$$

The final decision function,

$$f_t = \frac{1}{n} \sum_{i=1}^{n} \frac{exp\{\tau \cdot u_{ti}\}}{1 + exp\{\tau \cdot u_{ti}\}}, \ \tau > 0, \tag{3}$$

was calculated as a logistic average of single decision functions,

$$u_{ti} = \sum_{j=0}^{\ell} w_{ij} \cdot x_{tj},$$

where regression coefficients w were re-computed after feature reduction.

Remark 1. It is demonstrated in the Section 4 that performance of the classifier will be improved if we will use in (3) non-linear functions such as decision trees.

3 Boosting Algorithms

Boosting works by sequentially applying a classification algorithm to re-weighted versions of the training data, and then taking a weighted majority vote of the sequence of classifiers thus produced. For many classification algorithms, this simple strategy results in dramatic improvements in performance.

3.1 AdaBoost Algorithm

Let us consider minimizing the criterion [10]

$$\sum_{t=1}^{n} \xi(\mathbf{x}_t, y_t) \cdot e^{-y_t u(\mathbf{x}_t)}, \tag{4}$$

where the weight function is given below

$$\xi(\mathbf{x}_t, y_t) := \exp\{-y_t F(\mathbf{x}_t)\}. \tag{5}$$

We shall assume that the initial values of the ensemble d.f. $F(\mathbf{x}_t)$ are set to zero.

Advantages of the exponential compared with squared loss function were discussed in [9]. Unfortunately, we can not optimize the step-size in the case of exponential target function. We will need to maintain low value of the step-size in order to ensure stability of the gradient-based optimisation algorithm. As a consequence, the whole optimization process may be very slow and time-consuming. The AdaBoost algorithm was introduced in [12] in order to facilitate optimization process.

The following Taylor-approximation is valid under assumption that values of $u(\mathbf{x}_t)$ are small,

$$\exp\{-y_t u(\mathbf{x}_t)\} \approx \frac{1}{2}\left[(y_t - u(\mathbf{x}_t))^2 + 1\right]. \tag{6}$$

Therefore, we can apply quadratic-minimisation (QM) model in order to minimize (4). Then, we optimize value of the threshold parameter $\varDelta$ for u_t, and find the corresponding decision rule $f_t \in \{-1, 1\}$.

Next, we will return to (4),

$$\sum_{t=1}^{n} \xi(\mathbf{x}_t, y_t) \cdot e^{-c \cdot y_t \cdot f(\mathbf{x}_t)}, \tag{7}$$

where the optimal value of the parameter c may be easily found

$$c = \frac{1}{2} \log \{\frac{A}{B}\}, \tag{8}$$

and where

$$A = \sum_{y_t = f(\mathbf{x}_t)} \xi(\mathbf{x}_t, y_t), \quad B = \sum_{y_t \neq f(\mathbf{x}_t)} \xi(\mathbf{x}_t, y_t).$$

Finally (for the current boosting iteration), we update the function F:

$$F_{\text{new}}(\mathbf{x}_t) \leftarrow F(\mathbf{x}_t) + c \cdot f(\mathbf{x}_t), \tag{9}$$

and recompute weight coefficients ξ according to (5).

Remark 2. Considering test dataset (labels are not available), we will not be able to optimize value of the threshold parameter $\varDelta$. We can use either an average (predicted) value of $\varDelta$ in order to transform decision function into decision rule, or we can apply direct update:

$$F_{\text{new}}(\mathbf{x}_t) \leftarrow F(\mathbf{x}_t) + c \cdot u(\mathbf{x}_t), \tag{10}$$

where the value of the parameter $c \leq 1$ must be small enough in order to ensure stability of the algorithm.

3.2 LogitBoost Algorithm

Let us parameterize the binomial probabilities by

$$p(\mathbf{x}_t) = \frac{e^{2F(\mathbf{x}_t)}}{1 + e^{2F(\mathbf{x}_t)}}.$$

The binomial log-likelihood is

$$y_t^\star \log \{p(\mathbf{x}_t)\} + (1 - y_t^\star) \log \{1 - p(\mathbf{x}_t)\} = -\log \{1 + \exp \{-2 y_t F(\mathbf{x}_t)\}\}, \tag{11}$$

where $y^\star = (y + 1)/2$.

The following relation is valid,

$$\exp \{-2 y_t F(\mathbf{x}_t)\} = \xi(\mathbf{x}_t) z_t^2, \tag{12}$$

where

$$z_t = \frac{y_t^\star - p(\mathbf{x}_t)}{\xi(\mathbf{x}_t)}, \quad \xi(\mathbf{x}_t) = p(\mathbf{x}_t)(1 - p(\mathbf{x}_t)).$$

We can maximize (11) using a method with Newton's step, which is based on the matrix of second derivatives [11]. As an alternative, we can consider the standard weighted *QM*-model,

$$\sum_{t=1}^{n} \xi(\mathbf{x}_t)(z_t - u_t)^2. \tag{13}$$

After the solution $u(\mathbf{x}_t)$ was found, we update function $p(\mathbf{x}_t)$ as,

$$p(x_t) = \begin{cases} 1 & \text{if } h_t \geq 1; \\ h_t & \text{if } 0 < h_t < 1; \\ 0, & \text{otherwise,} \end{cases}$$

where $h_t = p(\mathbf{x}_t) + \xi(\mathbf{x}_t)u(\mathbf{x}_t)$. Then, we recompute the weight coefficients ξ and return to the minimization criterion (13).

Let us consider an update of the function F, assuming that $0 < h_t < 1$. By definition,

$$F_{\text{new}}(\mathbf{x}_t) = \frac{1}{2}\log\{\frac{h_t}{1 - h_t}\} = \frac{1}{2}\log\{\frac{p(\mathbf{x}_t)}{1 - p(\mathbf{x}_t)}\}$$

$$+\frac{1}{2}\log\{1 + \frac{u(\mathbf{x}_t)}{1 - p(\mathbf{x}_t)u(\mathbf{x}_t)}\} \approx F(\mathbf{x}_t) + \nu \cdot u(\mathbf{x}_t), \quad \nu = 0.5. \tag{14}$$

Remark 3. Boosting trick (similar to the well-known kernel trick): as an alternative to *QM*-solution, we can apply in (10) or (14) decision function, which was produced by another method, for example, Naïve Bayes, decision trees or random forest.

4 Experimental Results

4.1 Data Preparation

The given home-loan data includes two sets: 1) a training-set with 700 positive and 40000 negative instances, and 2) a test-set with 8000 instances. Any data-instance represents a vector of 40 continuous or categorical features. Using standard techniques, we reduced the categorical features to numerical (dummy) values. Also, we normalized the continuous values to lie in the range $[0, 1]$. As a result of the above transformation we created totally numerical dataset with $\ell = 101$ features. As a result of *MVF*, the number of features was reduced from 101 to 44 (see Figure 1).

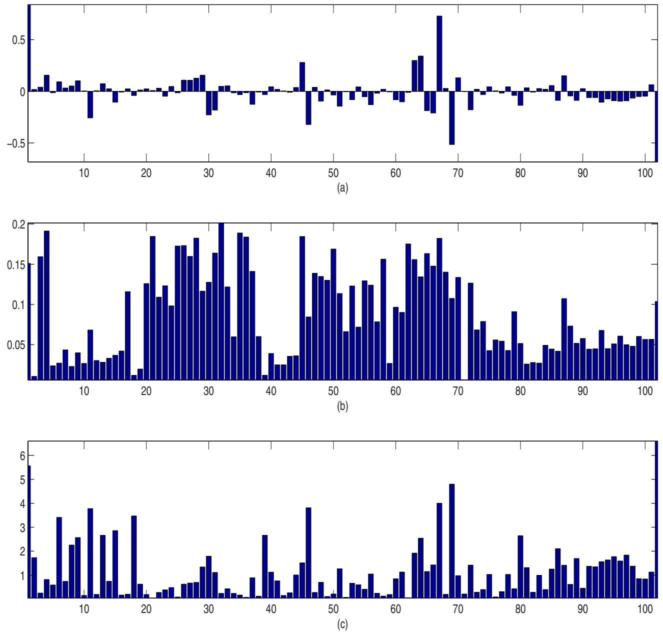

Fig. 1. Mean-variance filtering: (a) means (μ); (b) - standard deviations (s), (c) ratios $r = |\mu|/s$ (see Section 4 for more details)

Table 1. List of 6 the most significant features

N	Feature	μ	r
1	Bureau Enquiries for Morgages last 6 month	0.729	4
2	Age	-0.683	6.6
3	Bureau Enquiries for Loans last 12 month	-0.516	4.8
4	Bureau Enquiries last 3 month	0.342	2.54
5	Number of dependants	-0.322	3.82
6	Bureau Enquiries last month	0.299	1.92

4.2 Test Results

47 participants from various sources including academia, software vendors, and consultancies submitted entries with range of results from 0.4778 to 0.701 in terms of AUC. Our score was 0.688, which resulted in 9th place for us.

Note that the training AUC, which corresponds to the final submission was 0.7253. The difference between training and test results in the case of 44 features appears to be a quite significant. Initial thought (after results were published) was that there are problems with overfitting. We conducted series of experiments with feature selection, but did not make any significant improvement (Table 2).

Table 2. Numbers of the used features are given in the first column. Particular meanings of the features (in the cases of 4 and 17 features) may be found in the Tables 1, 3 and 4.

N of features	TestAUC	Base-learner
44	0.7023	LogitBoost
44	0.688	RLR
30	0.689	RLR
17	0.688	RLR
4	0.6558	RLR

Table 3. Top 4 features

N	Feature	μ	s	r
1	N1 (see Table 1)	1.1454	0.1011	11.3294
2	N3	-0.6015	0.0663	-9.0751
3	N5	-0.1587	0.0778	-2.0395
4	AGE	-0.6831	0.0806	-8.4794

As a next step, we decided to apply as a base-learner the ada-function in R. The best test result AUC = 0.7023 was obtained using the following settings: $loss = e, \nu = 0.3, type = gentle$.

We used in our experiment 100 random balanced sets. In addition, we conducted many experiments with up to 300 random sets, but we did not find any improvement. Also, it is interesting to note that we did not make any changes to the pre-processing technique, which was used before, and conducted our experiments against the same data.

4.3 Discussion and Business Insights

The RS-method provides good opportunities to evaluate the significance of the particular features. We can take into account 2 factors: 1) average values μ and 2) t-statistic r, which are defined in (2).

Based on the Tables 1 and 4, we can make a conclusion that younger people (AGE: $\mu = -0.654$) with smaller number of dependants (NBR OF DEPENDANTS: $\mu = -0.3298$) who made enquiries for mortgages during last 6 months have higher probability to take up a home loan.

On the other hand, enquiries for loans represent a detrimental factor ($\mu = -0.672$).

Considering a general characteristic such as marital status, we can conclude that "widowed" people are less interested ($\mu = -0.2754$) to apply for home loan.

Also, it is interesting to note that stable job (CURR EMPL MTHS: $\mu = -0.0288$) or long residence (CURR RES MTHS: $\mu = -0.0449$) may be viewed as negative factors. Possibly, these people have already one or more homes and are reluctant to make further investments.

Table 4. Top 17 features, which were selected using MVF

N	Feature	μ	s	r
1	MARITAL STATUS: married	0.0861	0.028	3.0723
2	MARITAL STATUS: single	0.0419	0.0236	1.7786
3	MARITAL STATUS: defacto	0.09	0.0438	2.0572
4	MARITAL STATUS: widowed	-0.2754	0.0766	3.594
5	RENT BUY CODE: mortgage	0.0609	0.0191	3.1838
6	RENT BUY CODE: parents	-0.1285	0.0341	3.7692
7	CURR RES MTHS	-0.0449	0.0101	4.4555
8	CURR EMPL MTHS	-0.0288	0.0111	2.586
9	NBR OF DEPENDANTS	-0.3298	0.0807	4.085
10	Bureau Enquiries last month	0.3245	0.183	1.7736
11	Bureau Enquiries last 3 month	0.1296	0.1338	0.9691
12	Bureau Enquiries for Morgages last 6 month	0.8696	0.1359	6.3982
13	Bureau Enquiries for Loans last 12 month	-0.6672	0.0795	8.3905
14	A DISTRICT APPLICANT=2	-0.1704	0.05	3.4067
15	A DISTRICT APPLICANT=8	-0.1216	0.0397	3.063
16	CUSTOMER SEGMENT=9	-0.0236	0.0317	0.7453
17	AGE	-0.654	0.0962	6.8015

Remark 4. Experiments with 'tree' function (R-software, package 'tree') had confirmed that the feature "Bureau enquiries for mortgages during last 6 month" is the most important.

With this model, the company can develop a marketing program such as a direct mail campaign to target customers with highest scores. For example, there are 350 positive cases in the independent test dataset with 8000 instances. We sorted the 8000 customers in a decreasing order according to the decision function with AUC = 0.7023 (see Table 2). As a result, we have found that 50%, 60% and 70% of all positive customers are contained in the field of 1770, 2519 and 3436 top scored customers.

4.4 Computation Time and Used Software

A Dell computer, Duo 2.6GHz, 3GB RAM, was used for computations. It took about 4 hours time in order to complete 100 balanced random sets and produce best reported solution.

5 Concluding Remarks and Further Developments

It is a well known fact that for various reasons it may not be possible to theoretically analyze a particular algorithm or to compute its performance in contrast to another. The results of the proper experimental evaluation are very important as these may provide the evidence that a method outperforms existing approaches. Data mining competitions are very important.

The proposed ensemble method is based on a large number of balanced random sets and includes 2 main steps: 1) feature selection and 2) training. During the PAKDD-2007 Data Mining Competition, we conducted both steps using linear regression. The proposed method is general and may be implemented in conjunction with different base-learners. In this paper we reported results which were obtained using the ADA package in R. These results outperform all known results.

Further improvement may be achieved as a result of more advanced preprocessing technique. Also, it appears to be promising to apply random forest as a single base-learner.

References

[1] Biau, G., Devroye, L., Lugosi, G.: Consistency of random forests and other averaging classifiers. Journal of Machine Learning Research 9, 2015–2033 (2007)

[2] Wang, W.: Some fundamental issues in ensemble methods. In: World Congress on Computational Intelligence, Hong Kong, pp. 2244–2251. IEEE, Los Alamitos (2008)

[3] Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)

[4] Zhang, B., Pham, T., Zhang, Y.: Bagging support vector machine for classification of SELDI-ToF mass spectra of ovarian cancer serum samples. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), vol. 4830, pp. 820–826. Springer, Heidelberg (2007)

[5] Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)

[6] Zhang, J., Li, G.: Overview of PAKDD Competition 2007. International Journal of Data Warehousing and Mining 4, 1–8 (2008)

[7] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, Heidelberg (2001)

[8] Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422 (2002)

[9] Nikulin, V.: Classification of imbalanced data with random sets and mean-variance filtering. International Journal of Data Warehousing and Mining 4, 63–78 (2008)

[10] Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Annals of Statistics 28, 337–374 (2000)

[11] Nikulin, V.: Learning with mean-variance filtering, SVM and gradient-based optimization. In: International Joint Conference on Neural Networks, Vancouver, BC, Canada, July 16-21, pp. 4195–4202. IEEE, Los Alamitos (2006)

[12] Freund, Y., Schapire, R.: A decision-theoretic generalization of online learning and an application to boosting. J. Comput. System Sciences 55, 119–139 (1997)