IOP Conference Series: Materials Science and Engineering

A New Improved Boosting for Imbalanced Data Classification

To cite this article: Zongtang Zhang *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **533** 012047

A New Improved Boosting for Imbalanced Data Classification

Zongtang Zhang, JiaXing Qiu and Weiguo Dai

Navy Submarine Academy, Qingdao, China

qtxy_robin@126.com, Qiujx8701@126.com, dwg1968@163.com

Abstract. As one of the most important component of artificial intelligence, machine learning is getting more and more attention. AdaBoost, a classic machine learning algorithm, is widely used. However, when faced with imbalanced data classification, AdaBoost's recognition rate of minority samples is low due to ignoring class imbalance. In many cases, minority samples are of high value. For this shortage, combining the theory of margin and cost-sensitive idea, a new Boosting algorithm called CMBoost is proposed based on cost-sensitive margin statistical characteristics, which is firstly through optimizing margin statistical characteristics to improve formal algorithm and then extended by cost-sensitive. Experimental results on the UCI dataset show that the CMBoost algorithm is superior to AdaBoost for imbalanced data classification problem.

1.Introduction

In recent years, imbalanced data classification has become a hot topic in the field of data mining and machine learning. Imbalanced dataset refer to the number of samples of one class in the dataset far less than the number of other classes, in which the majority of classes are called negative classes, while the minority is called positive class. The problem of classification of imbalanced dataset exists in people's real life and industrial production. For example, E-mail filtering[1], fraud detection[2], medical diagnosis[3], DNA microarray data analysis[4], software defect prediction[5], etc. In these applications, the classification accuracy of positive class is often more important. Therefore, to improve the classification accuracy of positive class has become the focus of research on imbalanced dataset.

AdaBoost[6] is a kind of strong learning algorithm to improve the weak learning algorithm performance, which has been applied to various fields of pattern recognition successfully[7,8,9]. The original AdaBoost algorithm treats both positive and negative samples equally and is not suitable for imbalanced data classification. Cost-sensitive AdaBoost introduces classification cost into AdaBoost algorithm, which can be used for imbalanced data classification. One of them is weighting method, which modifies the weight of Boosting update steps and is simple implementation, but lacks of theoretical support. Another one is the loss function method, which directly minimizes the expectation of cost sensitive loss. The algorithm is highly theoretical, but the realization is complex, which requires numerical method to solve.

In this paper, a new Boosting algorithm called CMBoost is proposed based on cost-sensitive margin statistical characteristics. Firstly, on the basis of generalization error bound based on the margin distribution, it is proposed to reduce the generalization error by optimizing the statistical characteristics of the margin, and then extend it with cost sensitivity, so that the algorithm can adapt to the imbalanced data. Experimental results on the UCI dataset show that the CMBoost algorithm is superior to the AdaBoost algorithm.

2. AdaBoost

The AdaBoost[10] algorithm was firstly proposed for bi-class classification. The algorithm takes as input a training set $\{(x_1,y_1),\cdots,(x_N,y_N)\}$ where with the sample x_i is an attribute value vector as a realization of the attribute set $X=\{X_1,X_2\cdots,X_N\}$, and class label y_i assumes a value in Y, with two classes, assuming that $Y=\{-1,1\}$. AdaBoost calls a given base learning algorithm repeatedly in a series of rounds $t=1,2,\cdots,M$. The weight of the i^{th} training sample on the iteration t is denoted by $D^t(i)$. Initially, all weights are set equally.

The base learner's task is to come up with a base classifier $h_t:X\to Y$ based on the distribution D^t to minimize the classification error. Once the base classifier h_t has been trained, AdaBoost chooses a parameter $\alpha_t\in R$ which measures the performance of the classification h_t. The data distribution D^t is then updated. The final classification criterion F is a weighted majority vote of the M base classifiers where α_t is the weight assigned to h_t.

3. CMBoost algorithm

We use $\Pr_D[\cdot]$ to refer as the probability with respect to D and $E[\cdot]$ to denote the expected values. Let H be a base classifier space and $C(H)$ denote the convex hull of H. A voting classifier $f\in C(H)$ is of the following form

$$f=\sum\alpha_i h_i \quad with \sum\alpha_i=1 \quad and \ \alpha_i\geq 0 \tag{1}$$

For an example (x,y), the margin with respect to the voting classifier $f=\sum\alpha_i h_i$ is defined as $yf(x)$ or ρ, in other words

$$yf(x)=\rho=\sum_{i:y=h_i(x)}\alpha_i-\sum_{i:y\neq h_i(x)}\alpha_i \tag{2}$$

Schapire[11]derived the generalization error bounds based on the margin distribution, as shown in Theorem 1

Theorem 1: D_{tr} is made up of N training samples which are selected randomly from a distribution $Dist$ on the training set $X\times Y$. Assuming that the base classifier space H is limited and $\delta>0$, for all $\theta>0$, every voting classifier $f\in C(H)$ satisfies the following bound with probability at least $1-\delta$

$$\begin{aligned}\Pr_{Dist}[yf(x)<0]&\leq\Pr_{D_{tr}}[yf(x)\leq\theta]\\&+O\Big(\frac{1}{\sqrt{N}}\Big(\frac{\ln N\ln|H|}{\theta^2}+\ln\frac{1}{\delta}\Big)^{1/2}\Big)\end{aligned} \tag{3}$$

As we can see, the generalization error bounds relate to the margin distribution, but how do we get a good margin distribution? We know that statistical characteristics are good tools for describing distributions. Therefore, on the basis of bounds of margin in the past, the generalization error bounds derived from literature[12] reveal the relationship between them and the statistical characteristics of margin, as shown in Theorem 2:

Theorem 2: D_{tr} is made up of $N(N>5)$ training samples which are selected randomly from a distribution $Dist$ on the training set $X\times Y$. For all $\delta>0$, every voting classifier $f\in C(H)$ satisfies the following bound with probability at least $1-\delta$:

$$\begin{aligned}\Pr_{Dist}[yf(x)<0]\leq\frac{1}{N^{50}}+\inf_{\theta\in(0,1]}\{&\Pr_{Dtr}[yf(x)<\theta]\\&+N^{-2/(1-E_{Dtr}^2[yf(x)]+\theta/9)}\\&+\frac{3\sqrt{\mu}}{N^{3/2}}+\frac{7\mu}{3N}+\sqrt{\frac{3\mu}{N}\Gamma(\theta)}\}\end{aligned} \tag{4}$$

where

$$\begin{aligned} &\mu = 144\ln m \ln(2|H|)/\theta^2 + \ln(2|H|/\delta) \\ &\Gamma(\theta) = \Pr_{D_{tr}}[yf(x) < \theta] \Pr_{D_{tr}}[yf(x) \geq 2\theta/3] \end{aligned} \tag{5}$$

In the theorem, $E[yf(x)]$ is the margin mean and $\Gamma(\theta)$ reflects the size of the margin variance. It can be seen that the margin statistical characteristics of training sample set directly affect the size of generalization error. When the size of the training sample set and the complexity of the base classifier are fixed, if the margin mean gets larger and the margin variance gets smaller at the same time, the generalization error will get smaller.

Therefore, the algorithm performance is improved by maximizing the margin mean and minimizing the margin variance. If mg_{ave} is the margin mean, mg_{var} is the margin variance and $\lambda_1 > 0, \lambda_2 > 0$ are weighing coefficient, we can construct the margin statistical characteristics

$$mg_{sc} = \lambda_1 mg_{\text{var}} - \lambda_2 mg_{ave} \tag{6}$$

Then the improved AdaBoost algorithm is represented by the following formula:

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \ mg_{sc} \\ &s.t. \ \ \boldsymbol{\alpha} \geq 0, 1^T \boldsymbol{\alpha} = D \end{aligned} \tag{7}$$

where

$$mg_{ave} = \frac{1}{N}\sum_{i=1}^{N} \rho_i \tag{8}$$

$$mg_{\text{var}} = \frac{1}{N-1}\sum_{i>j} (\rho_i - \rho_j)^2 \tag{9}$$

We have

$$\begin{aligned} mg_{sc} &= \lambda_1 mg_{\text{var}} - \lambda_2 mg_{ave} \\ &= \lambda_1 \frac{1}{N-1}\sum_{i>j}(\rho_i - \rho_j)^2 - \lambda_2 \frac{1}{N}\sum_{i=1}^{N}\rho_i \\ &= 2\lambda_1 [\frac{1}{2(N-1)}\sum_{i>j}(\rho_i - \rho_j)^2 - \frac{\lambda_2}{2N\lambda_1}\sum_{i=1}^{N}\rho_i] \end{aligned} \tag{10}$$

Defining

$$A = \begin{bmatrix} 1 & c & \cdots & c \\ c & 1 & \cdots & c \\ \vdots & \vdots & \ddots & \vdots \\ c & c & \cdots & 1 \end{bmatrix}_{N\times N}, c = \frac{-1}{N-1} \tag{11}$$

$$\boldsymbol{\lambda} = -\frac{\lambda_2}{2N\lambda_1} I_{N\times 1} \tag{12}$$

$$\boldsymbol{\rho} = [\rho_1; \rho_2; \cdots; \rho_N] \tag{13}$$

We get

$$mg_{sc} = 2\lambda_1(\frac{1}{2}\boldsymbol{\rho}^T A \boldsymbol{\rho} + \boldsymbol{\lambda}^T \boldsymbol{\rho}) \tag{14}$$

Then we can rewrite (7) as

$$\begin{aligned} &\min_{\boldsymbol{\alpha},\boldsymbol{\rho}} \frac{1}{2}\boldsymbol{\rho}^T A \boldsymbol{\rho} + \boldsymbol{\lambda}^T \boldsymbol{\rho} \\ &s.t. \ \ \boldsymbol{\alpha} \geq 0, 1^T \boldsymbol{\alpha} = D \\ &\qquad \rho_i = y_i \boldsymbol{h}_i^T \boldsymbol{\alpha}, \forall i = 1, 2, \cdots, N. \end{aligned} \tag{15}$$

where

$$\boldsymbol{h}_i = [h_1(x_i); h_2(x_i); \cdots; h_M(x_i)] \tag{16}$$

$$\boldsymbol{\alpha} = [\alpha_1; \alpha_2; \cdots; \alpha_M] \tag{17}$$

As $\boldsymbol{x}^T A \boldsymbol{x} \geq \boldsymbol{0}$ is always holds for any real non-zero vector $\boldsymbol{x}$, so A is positive semidefinite matrix.

In the training sample set, the size of the positive sample set L_+ is n_+ and the size of the negative sample set L_- is n_-, then the imbalance degree of the sample set is $r_u = n_- / n_+$. Suppose the weight of each sample when calculating the margin mean is d_i, the weight when calculating the margin variance is k_i and the cost sensitive parameter is c, then

$$d_i = \begin{cases} 1 + cr_u \left|\ln r_u\right|, i \in L_+ \\ 1 + c\dfrac{1}{r_u}\left|\ln \dfrac{1}{r_u}\right|, i \in L_- \end{cases} \tag{18}$$

$$k_i = \begin{cases} 1 + c\dfrac{1}{r_u}\left|\ln \dfrac{1}{r_u}\right|, i \in L_+ \\ 1 + cr_u \left|\ln r_u\right|, i \in L_- \end{cases} \tag{19}$$

We have cost-sensitive margin mean

$$cmg_{ave} = \frac{1}{N}\sum_{i=1}^{N} d_i \rho_i \tag{20}$$

and cost-sensitive margin variance

$$cmg_{\mathrm{var}} = \frac{1}{N-1}\sum_{i>j}(k_i\rho_i - k_j\rho_j)^2 \tag{21}$$

Then we can construct the cost-sensitive margin statistical characteristics:

$$\begin{aligned} cmg_{sc} &= \lambda_1 cmg_{\mathrm{var}} - \lambda_2 cmg_{ave} \\ &= 2\lambda_1[\frac{1}{2(N-1)}\sum_{i>j}(k_i\rho_i - k_j\rho_j)^2 \\ &\quad - \frac{\lambda_2}{2N\lambda_1}\sum_{i=1}^{N} d_i\rho_i] \end{aligned} \tag{22}$$

Defining

$$K = diag(k_1, k_2, \cdots, k_N) \tag{23}$$

$$B = K^T A K \tag{24}$$

$$\boldsymbol{d} = [d_1, d_2, \cdots, d_N] \tag{25}$$

$$\boldsymbol{\lambda}_g = -\frac{\lambda_2}{2N\lambda_1}\boldsymbol{d} \tag{26}$$

Then

$$cmg_{sc} = 2\lambda_1(\frac{1}{2}\boldsymbol{\rho}^T B\boldsymbol{\rho} + \boldsymbol{\lambda}_g^T\boldsymbol{\rho}) \tag{27}$$

Then the CMBoost algorithm is represented by the following formula:

$$\begin{aligned} &\min_{\alpha,\rho} \frac{1}{2}\boldsymbol{\rho}^T B\boldsymbol{\rho} + \boldsymbol{\lambda}_g^T\boldsymbol{\rho} \\ &s.t. \quad \boldsymbol{\alpha} \geq 0, 1^T\boldsymbol{\alpha} = D \\ &\qquad \rho_i = y_i\boldsymbol{h}_i^T\boldsymbol{\alpha}, \forall i = 1, 2, \cdots, N. \end{aligned} \tag{28}$$

As $\boldsymbol{x}^T B\boldsymbol{x} \geq \boldsymbol{0}$ is always holds for $\boldsymbol{x}^T A\boldsymbol{x} \geq \boldsymbol{0}$, then B is positive semidefinite matrix. So (28) is a convex quadratic problem.

Because the H is unknown, we can't solve the problem by usual QP solvers. As in LPBoost[13], column generation can be used to attack this problem. Then we apply column generation to this problem.

The Lagrangian of (28) is

$$\begin{aligned} L(\boldsymbol{\alpha},\boldsymbol{\rho},\boldsymbol{u},r,\boldsymbol{q}) &= \frac{1}{2}\boldsymbol{\rho}^T B\boldsymbol{\rho} + \boldsymbol{\lambda}_g{}^T\boldsymbol{\rho} + r(\mathbf{1}^T\boldsymbol{\alpha} - D) \\ &\quad - \boldsymbol{q}^T\boldsymbol{\alpha} + \sum_{i=1}^{N} u_i[\rho_i - y_i\boldsymbol{h}_i^T\boldsymbol{\alpha}] \end{aligned} \tag{29}$$

The infimum of L is

$$\begin{aligned} \inf_{\boldsymbol{\rho},\boldsymbol{\alpha}} L &= \inf_{\boldsymbol{\rho}}[\frac{1}{2}\boldsymbol{\rho}^T B\boldsymbol{\rho} + (\boldsymbol{u}+\boldsymbol{\lambda}_g)^T\boldsymbol{\rho}] \\ &\quad + \inf_{\boldsymbol{\alpha}}[(r\mathbf{1}^T - \boldsymbol{q}^T - \sum_{i=1}^{N} u_i y_i \boldsymbol{h}_i^T)\boldsymbol{\alpha}] - Dr \end{aligned} \tag{30}$$

For finite infimum, $r\mathbf{1}^T - \boldsymbol{q}^T - \sum_{i=1}^{N} u_i y_i \boldsymbol{h}_i^T = \mathbf{0}$ must hold. So we have

$$\sum_{i=1}^{N} u_i y_i \boldsymbol{h}_i^T \leq r\mathbf{1}^T \tag{31}$$

For the first term in L

$$\frac{\partial[\frac{1}{2}\boldsymbol{\rho}^T B\boldsymbol{\rho} + (\boldsymbol{u}+\boldsymbol{\lambda}_g)^T\boldsymbol{\rho}]}{\partial\rho_i} = 0, \forall i \tag{32}$$

This results in $\boldsymbol{\rho} = -B^{-1}(\boldsymbol{u}+\boldsymbol{\lambda}_g)$, and the infimum is $-\frac{1}{2}(\boldsymbol{u}+\boldsymbol{\lambda}_g)^T B^{-1}(\boldsymbol{u}+\boldsymbol{\lambda}_g)$.

Through putting the results together, the dual is

$$\begin{aligned} &\max_{r,\boldsymbol{u}} -\frac{1}{2}(\boldsymbol{u}+\boldsymbol{\lambda}_g)^T B^{-1}(\boldsymbol{u}+\boldsymbol{\lambda}_g) - Dr \\ &s.t. \sum_{i=1}^{N} u_i y_i \boldsymbol{h}_i^T \leq r\mathbf{1}^T \end{aligned} \tag{33}$$

We can reformulate (28) as

$$\begin{aligned} &\min_{r,\boldsymbol{u}} \frac{1}{2D}(\boldsymbol{u}+\boldsymbol{\lambda}_g)^T B^{-1}(\boldsymbol{u}+\boldsymbol{\lambda}_g) + r, \\ &s.t. \sum_{i=1}^{N} u_i y_i \boldsymbol{h}_i^T \leq r\mathbf{1}^T \end{aligned} \tag{34}$$

To speed up the convergence, we add the most violated constraint by solving the following problem:

$$h^{'}(\cdot) = \arg\max_{h(\cdot)} \sum_{i=1}^{N} u_i y_i h(x_i) \tag{35}$$

4. Evaluation method

In traditional classification learning, classification accuracy is generally used as the evaluation criterion. However, for imbalanced dataset, it is unreasonable to use classification accuracy to evaluate the performance of classifiers. Therefore, for imbalanced data, many scholars propose evaluation methods such as F-measure[14] and G-mean[15], most of which are based on the confusion matrix (see Table 1). In Table 1, TP and TN respectively represent the number of positive and negative samples of the correct classification, while FP and FN respectively represent the number of positive and negative samples of the wrong classification.

Table 1. Confusion matrix for bi-class problem

	Be classified into positive class	Be classified into negative class
True positive class	TP	FN
True negative class	FP	TN

F-measure is an evaluation criterion for imbalanced data classification problem, which is defined as follows

$$F-measure=\frac{(1+\beta^2)\times Recall\times Precision}{\beta^2\times Recall+Precision} \tag{36}$$

$$Recall=\frac{TP}{TP+FN},\ Precision=\frac{TP}{TP+FP} \tag{37}$$

Only when the recall and precision are both large, F-measure will be large. Therefore, it is reasonable to evaluate the classification performance of classifiers for positive class.

G-mean is an evaluation criterion for the overall classification performance , which is defined as follows:

$$G-mean=\sqrt{Positive\ Accurary\times Negative\ Accurary} \tag{38}$$

$$Positive\ Accurary=Recall=\frac{TP}{TP+FN} \tag{39}$$

$$Negative\ Accurary=\frac{TN}{TN+FP} \tag{40}$$

G-mean is to maximize the accuracy of two classes while maintaining the balance of the classification accuracy of positive and negative class. In this article, F-measure is used to evaluated the classification performance of the positive class, while G-mean is used to evaluated the overall classification performance.

5. Experiment and analysis

To test the effectiveness of the proposed method, several UCI dataset are selected. For data containing multiple classes, one of them is taken as positive and the rest as negative. The information of UCI dataset is given in Table 2. The dataset is divided into two halves. One half is training sample set and the other is test sample set. F-measure and G-mean are used to evaluate the performance of AdaBoost and CMBoost which is shown in Table 3.

Table 2. Information of UCI dataset

Dataset	Total size	Positive class size	Negative class size	Imbalance degree
heart	303	139	164	1.18
sonar	208	97	111	1.14
vehicle	846	212	634	2.99
wine	178	59	119	2.02
wpbc	198	47	151	3.21
segment	2310	330	1980	6.00
vote	435	168	267	1.59

Table 3. Algorithm performance comparison

Dataset	Algorithm	F-measure	G-mean
heart	AdaBoost	0.78	0.81
	CMBoost	0.82	0.83
sonar	AdaBoost	0.74	0.76
	CMBoost	0.78	0.82
vehicle	AdaBoost	0.26	0.41
	CMBoost	0.42	0.68
wine	AdaBoost	0.94	0.96
	CMBoost	0.94	0.95
wpbc	AdaBoost	0.77	0.44
	CMBoost	0.86	0.59
segment	AdaBoost	0.43	0.74
	CMBoost	0.58	0.86
vote	AdaBoost	0.93	0.95
	CMBoost	0.94	0.95

From Table 3, first of all, we can see that the mean F-measure of CMBoost is 7% larger than that of AdaBoost , while the mean G-mean of CMBoost is 8.8% larger than that of AdaBoost. Besides, for almost balanced dataset like heart, sonar, vote, AdaBoost performs very well. While for imbalanced dataset like vehicle, segment, the performance of AdaBoost is bad. The CMBoost has more enhancements on the imbalanced dataset than the balanced dataset.

6. Conclusion

Many domains are affected by the problem of class imbalance. It can be challenging to construct a classifier that effectively identifies the examples of the positive class. Several techniques have been proposed for dealing with the problem of class imbalance. In this paper, we have proposed a new algorithm, called CMBoost, for alleviating the problem of class imbalance. We compare the performance of CMBoost with AdaBoost from UCI dataset, finding that CMBoost performs better when compared to AdaBoost, especially for imbalanced dataset.

References

[1]Dai HL. Class imbalance learning via a fuuzy total margin based support vector machine. Applied Soft Computing, 31: 172-184, 2015

[2]Deng X, Tian X. Nonlinear process fault pattern recognition using statistics kernel PCA similarity factor. Neurocomputing, 121(18):298-308, 2013

[3]Ozcift A, Gulten A. Classifer ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms. Computer Methods Programs Biomedicine, 104(3):443-451, 2011

[4]Yu H, Ni J, Zhao J. ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data. Neurocomputing, 101:309-318, 2013

[5]Wang S, Yao X. Using class imbalance learning for software defect prediction. IEEE Trans on Reliability, 62(2):434-443, 2013

[6]Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to Boosting[J]. Journal of Computer and System Sciences, 55(1): 119-139, 1997

[7]Peng X J, Setlur S, Govindaraju V, Ramachandrula S. Using a boosted tree classifer for text segmentation in handannotated documents. Pattern Recognition Letters, 33(7): 943-950, 2012

[8]Ren J F, Jiang X D, Yuan J S. A complete and fully automated face verification system on mobile devices. Pattern Recognition, 46(1): 45-56, 2013

[9]Ren S K, Hou Y X, Zhang P, Liang X R. Importance weighted AdaRank. Proceedings of the 7th International Conference on Advanced Intelligent Computing. Berlin, Heidelberg: Springer-Verlag, 6838: 448-455, 2012

[10]Sun Y M, Kamel M S, Wong A K C, Wang Y. Cost-sensitive Boosting for classification of imbalanced data. Pattern Recognition, 40(12): 3358-3378, 2007

[11]RE Schapire, Y Freund, P Bartlett, WS Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. Annals of Statistics, , 26(5): 1651-1686 1998

[12]Wei Gao, Zhihua Zhou. On the doubt about margin explanation of boosting. Artificial Intelligence, 203:1-18, 2013

[13]A. Demiriz, K.P. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. Mach. Learn., 46225–254, 2002

[14]Han Hui, Wang Wenyuan, Mao Binghuan. Borderline-SMOTE：A New Over-Sampling Method in imbalanced Data Sets Learning. Pine of the International Conference on Intelligent Computing: 878—887, 2005

[15]SU C T, Chen hongsheng, Yih Y.Knowledge Acquisition through Information Granulation for lmbalanced Data． Expert Systems Tools and Applications, 31(3):531—541, 2006