A Comparison of Boosting Tree and Gradient Treeboost Methods for Carpal Tunnel Syndrome

Karpal Tunel Sendromu İçin Boosting Tree ve Gradient Treeboost Algoritmalarının Karşılaştırılması

Gülhan OREKİCİ TEMEL,[a]
Handan ANKARALI,[b]
Bahar TAŞDELEN,[a]
Semra ERDOĞAN,[a]
Aynur ÖZGE[c]

Departments of
[a]Biostatistics and Bioinformatics,
[c]Neurology,
Mersin University Faculty of Medicine,
Mersin
[b]Department of Biostatisticsand Bioinformatics,
Düzce University Faculty of Medicine,
Düzce

Geliş Tarihi/*Received:* 27.11.2013
Kabul Tarihi/*Accepted:* 27.02.2014

Yazışma Adresi/*Correspondence:*
Gülhan OREKİCİ TEMEL
Mersin University Faculty of Medicine,
Department of Biostatistics and Bioinformatics, Mersin,
TÜRKİYE/TURKEY
gulhan_orekici@hotmail.com

ABSTRACT Objective: Boosting is one of the most successful combining methods. The principal aim of these combining algorithms is to obtain a strong classifier with small estimation error from the combination of weak classifiers. Boosting based on combining tree has many advantages. Data sets can contain mixtures of nominal, ordinal and numerical variables. AdaBoost and Gradient TreeBoost are commonly used boosting procedure. Both methods are a stage wise additive model fitting procedure. Our goal in this study is to explain the both method and to compare the algorithm results on a neurology data set on the purpose of classification. **Material and Methods:** The data set consists of 4076 incidences in total. The condition of being a patient with Carpal Tunnel Syndrome (CTS) or not was considered as the dependent variable. Boosting Tree and Gradient TreeBoost applications were conducted in Statistica 7.0 and Salford Predictive Modeler: TreeNet (R) trial version 6.6.0.091. **Results:** In AdaBoost and Gradient TreeBoost algorithm, multiple trees are grown of the training data. 200 trees are produced for both models. 70 trees in the AdaBoost Algorithm and 196 trees in the Gradient TreeBoost algorithm are chosen as the optimal trees. **Conclusion:** The sensitivity or specify values in the test data of Gradient TreeBoost are high indicates that they can be used as a successful method in CTS diagnosis. . It is believed that the boosting methods will become very more and more popular in health science due to its easy implementation and high predictive performance.

Key Words: Boosting; AdaBoost; Gradient Boosting; Additive Models

ÖZET Amaç: Boosting algoritması en başarılı birleştirme algoritmasıdır. Birleştirme algoritmalarının temel amacı, zayıf sınıflayıcıların birleşiminden, tahmin hatası daha düşük güçlü sınıflayıcılar elde etmektir. Boosting temelli oluşturulan birleştirilmiş ağaçların birçok avantajları vardır. Bu tür ağaç modelleri kategorik, sıralı ya da sürekli yapıda değişkenler için kullanılabildiği gibi bu değişkenlerin karma yapısında da boosting temelli algoritmalar kullanılabilir. AdaBoost ve Gradient TreeBoost boosting algoritmalarında en yaygın kullanılan algoritmalardır. Her iki metotta eklemeli model prensibini kullanır. Bizim bu çalışmadaki amacımız bu iki metodu açıklamak ve bir nöroloji veri seti üzerinde yöntemleri sınıflama amaçlı karşılaştırmaktır. **Gereç ve Yöntemler:** Veri setinde toplam 4076 vaka vardır. Bağımlı değişken olarak da vakaların Karpal Tunel Sendromu (KTS) olup olmama durumları alınmıştır. Boosting Tree algoritmasında Statistica 7.0 paket programı ve Gradient TreeBoost'ta ise Salford Predictive Modeler: TreeNet (R) deneme sürümü 6.6.0.091 kullanılmıştır. **Bulgular:** AdaBoost ve Gradient TreeBoost ile çok sayıda ağaç oluşturabilir. Her iki model için 200 ağaç oluşturulmuştur. AdaBoost algoritması için 70, Gradient TreeBoost algoritması için 196 ağaç en başarılı ağaç olarak seçilmiştir. **Sonuç:** KTS tanısını koymada kullanılan metotların her ikisi de başarılır. Gradient TreeBoost algoritmasının test verisinde de duyarlılık ve seçicilik değeri AdaBoost algoritmasına göre daha yüksektir. Boosting temelli modellerin tahmin başarısının yüksek olması ve yorumlanmasının kolaylığından kaynaklı olarak her geçen gün sağlık bilimlerinde daha popüler olacağı düşünülmektedir.

Anahtar Kelimeler: Boosting; AdaBoost; Gradient Boosting; Eklemeli modeller

Turkiye Klinikleri J Biostat 2014;6(2):67-73

Boosting is an ensemble learning method for improving the predictive performance of classification or regression procedures, such as decision trees.[1] Boosting is one of the most successful combining methods. The principal aim of these combining algorithms is to obtain a strong classifier with small estimation error from the combination of weak classifiers.[2] The methods combine re-weighted weak classifiers linearly to find the strong classifier. The boosting algorithm was developed in 1990 by Schapire.[2] In modeling of Classification and Regression Tree is obtained to better performance, there are actually several implementations of boosting, such as AdaBoost,[1,3] Friedman's gradient boosting,[4] stochastic variants of boosting[5] and many others.[6,7] AdaBoost, as the most popular boosting procedure, was introduced by Freund and Schapire in 1995 and also extended to multi-class problems.[6,8,9] Moreover, a different boosting procedure put forth in 1999 by Friedman known for the Gradient TreeBoost is in the literature.[10] Both methods are stage wise additive models fitting procedure that can enhance the predictive performance of learning algorithms. These methods provide solution to both classification and regression problems by using different loss functions. There are a great number of studies in the literature about the successful estimation of the boosting models.[10-12]

Boosting method can be used for both categorical and continuous dependent variables. There is no limitation about the distribution of independent variables. They can be continuous, categorical or mixed type of distributions. Hundreds of different TreeBoost models can be formed with the current independent variables. It can remove the irrelevant and insignificant independent variables automatically from the estimation models. Missing values and outlier do not pose a problem in the model.

Boosting is a bootstrap-based method, all observations are being moved by chance and they are used in modeling. Trees built by this way resist over fitting, since the boosting both reduces the training classification error and maximizes the classification margin separating the two classes.[13,14] Boosting method is generally performs better to other commonly used methods.[10-12,15]

Hundreds of trees can be formed in boosting-based tree models. While other decision tree models are total parallel single trees, the boosting-based tree models cannot be monitored with a single tree. Therefore, this model is complex. In this situation, this can be considered as a disadvantage of this model.[16] Our goal in this study is to explain the AdaBoost and Gradient TreeBoost algorithms and to compare the algorithm results on a neurology data set on the purpose of classification.

MATERIAL AND METHODS

ADABOOST ALGORITHM

The Boosting, one of the most successful combining methods, was proposed by Schapire.[2] The most popular algorithm AdaBoost was introduced by Freund and Schapire in 1995 and also extended to multi-class problems.[6,8,9] This algorithm was called as Real AdaBoost for two-class dependent variable and AdaBoost.MH for more than two classes.[10] The weak classifiers used in AdaBoost algorithm are single-split classification trees.[17] In Boosting, a sequence of trees is obtained from reweighting data after each classification tree. In each stage of boosting the weight of wrongly classified patients are increased while the weight of correctly classified patients is decreased.[18]

Figure 1 shows a schematic of the AdaBoost procedure.[10] The process of averaging weighted classifiers not only reduces the fitting error rate but also protects against over fitting.[10] The systematic process of AdaBoost algorithm starts with uniform

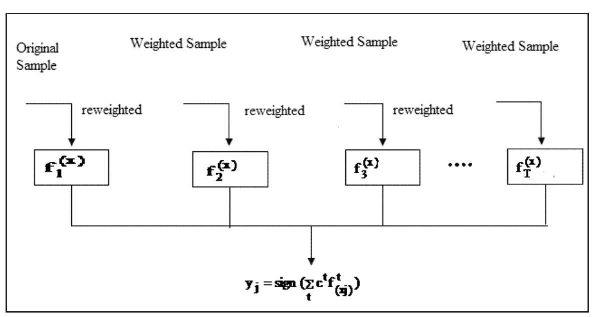

FIGURE 1: The schematic of the AdaBoost procedure.

distribution of weights over training samples of patients. Using a classifier f(x) and confidence index, each case is classified initially. Increasing the weights on misclassified patients, each case is reclassified. The process is repeated until convergence via a sign function is used for final decision.[10]

Let the training set with n observations (i=1,2,...,n) for p independent variables be given by:

$$D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\} \quad (1)$$

Where $x_i = (x_{1i}, .., x_{pi})$ and $y \in \{+1, -1\}$. (2)

AdaBoost algorithm for such sample includes following steps:

Step1. Each subject in the training set is weighted equally.

$$(w_i^1 = \frac{1}{N}) \quad (i = 1,2,..., N) \quad (3)$$

Step2. For each iteration (t= 1,2,...,T)

a. A data set with n subject is resembled with the Bootstrap technique. Sampling probability of high weighted subject is more than the others.

b. A classifier f(x) is obtained using CART technique.[19]

c. An indicator function is described to calculate classification error rate of f(x). The function for each iteration is given in following way. If a sample is misclassified 1, otherwise zero

$$err_i^t = \begin{Bmatrix} I(y(i)) = f(x_i) = 0 \\ I(y(i)) \neq f(x_i) = 1 \end{Bmatrix} \quad (4)$$

When a subject is wrong classified, $err_i^t = 1$, otherwise $err_i^t = 0$.

d. After the calculation of err_i^t for each iteration, weighted sum of all training set errors and confidence index (c^t) for f(x) classifier is calculated.[20-22]

$$err^t = \sum_i (w_i^t err_i^t) \quad (5)$$

$$c^t = \log(\frac{(1 - err^t)}{err^t}) \quad (6)$$

The lower the weighted errors are the higher confidence index will be.

e. All training sets are reweighted provided that $\sum_i w_i^{t+1} = 1$.

$$w_i^{t+1} = w_i^t \exp(c^t err_i^t) \qquad i = 1,2,..., N \quad (7)$$

f. If $err_i^t \leq 0.5$ and t<T (t=t+1), steps (a)-(f) are repeated, otherwise the iteration is stopped.

Step3. The performance of discrete AdaBoost algorithm is calculated using a test set. The final estimation for a sample in test set is combination of estimations from T classifiers.

$$y_j = \text{sign}(\sum_t c^t f^t_{(xj)}) \quad (8)$$

Where sign function is used to estimate dependent variable. [10]

$$sign(*) = \begin{vmatrix} -1, * < 0 \\ 1, * \geq 0 \end{vmatrix} \quad (9)$$

GRADIENT TREEBOOST ALGORITHM

The Gradient TreeBoost Algorithm is developed by Friedman in 1999.[4] This method has different names in the literature. The algorithm is functionally similar to the decision tree. The tree formed here is a tree community. Trees involve generating a sequence of trees, each grown on the residuals of the previous tree. Figure 2 shows a schematic of the Gradient TreeBoost procedure.

The Gradient TreeBoost model is mathematically defined as in Equation 10.

$$F(x) = F_0 + \beta_1 h_1(X) + \beta_2 h_2(X) + ... + \beta_M h_M(X) \quad (10)$$

In a data set, in which $x = \{x_1, ..., x_n\}$ independent variable and $y \rightarrow$ dependent variable are defined, it should be defined as $\{y_i, x_i\}_1^N$ as a training set from (y, x) known values set. Here, h_i is a small tree. The model is composed of hundreds of small

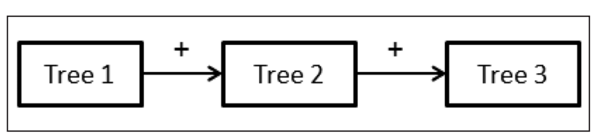

FIGURE 2: The schematic of the Gradient TreeBoost procedure.

trees. The first tree is formed from the data. Then the mistakes obtained from the first tree are used to decrease the mistakes in the second tree; the second tree utilizes the first tree. The process is repeated until a set is formed of the successful trees. Each tree is more homogenous than the preceding one. The model bases technically on an iteration algorithm. The goal here is to find the $F^*(x)$ function with the help of known values. This process is carried out by minimizing the expected values of the $\Psi(y, F(x))$ loss function.[4,5]

$$F^*(x) = \arg\min_{F(x)} E_{y,x}\Psi(y, F(x)) \qquad (11)$$

In the boosting approach, an additive function is defined for $F^*(x)$.

$$F(x) = \sum_{m=0}^{M} \beta_m h(x; a_m) \qquad (12)$$

Here, $h(x;a)$ is a base learner. The base learner x independent variable is determined with $a = \{a_1, a_2, ...\}$ parameters. The $\{\beta_m\}_0^M$ coefficient and $\{a_m\}_0^m$ parameter are obtained from the training data. The algorithm starts with $F_0(x)$ and continues until m=1,2,…,M.[4,5]

$$(\beta_m, a_m) = \arg\min_{\beta,a} \sum_{i=1}^{N} \Psi(y_i, F_{m-1}(x_i) + \beta h(x_i; a)) \qquad (13)$$

and

$$F_m(x) = F_{m-1}(x) + \beta_m h(x; a_m) \qquad (14)$$

According to the Gradient Boosting approach, the $\Psi(y, F(x))$ loss function in Equation 11 is analyzed in two phases.[5]

The First Phase

$h(x;a)$ function is calculated with the least squares technique to calculate the pseudo-residuals obtained from the dependent variable values in response to the independent variable values in that node.[5]

$$(a_m) = \arg\min_{a,\rho} \sum_{i=1}^{N} [\tilde{y}_{im} - \rho h(x_i; a)]^2 \qquad (15)$$

ρ is a scaler. $\rho \in IR^N$

$$\tilde{y}_{im} = -\left[\frac{\partial \Psi(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}^{(x)}} \qquad (16)$$

The Second Phase

It is calculation of the optimal β_m coefficient for the $h(x; a_m)$ function.

$$\beta_m = \arg\min_{\beta} \sum_{i=1}^{N} \Psi(y_i, F_{m-1}(x_i) + \beta h(x_i; a_m)) \qquad (17)$$

In this approach, the base learner is a regression tree having an $h(x;a)$ L unit terminal node. In each m. iteration, a regression tree separates the x independent variable space into L terminal node (region) $\{R_{lm}\}_{l=1}^{L}$. It estimates a constant value in each node.[4,5]

$$h(x; \{R_{jm}\}_1^J = \sum_{l=1}^{L} \bar{y}_{lm} I(x \in R_{lm}) \qquad (18)$$

Here, it is an average of the pseudo-residuals obtained from the y values corresponding to x independent variable value in the R_{lm} terminal node in $\bar{y}_{lm} = \text{mean}_{x_i \in R_{lm}} (\tilde{y}_{im})$ m. iteration.[4,5]

$$\gamma_{lm} = \arg\min_{\gamma} \sum_{x_i \in R_{lm}} \Psi(y_i, F_{m-1}(x_i) + \gamma) \qquad (19)$$

Then,

$F_{m-1}(x)$ is updated in each terminal node.

$$F_m(x) = F_{m-1}(x) + v.\gamma_{lm} I(x \in R_{lm}) \qquad (20)$$

Each constructed model is shrunk by a shrinkage parameter v, which is a positive value than 1 (0<v≤1). The v acts as a weight for current model which prevents possible over fitting by constraining the fitting process.[4,5]

The generalized boosting decision tree algorithm is similar to the one in Table 1.[1,4,5,16]

TABLE 1: Gradient TreeBoost.

1	$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{N} \Psi(y_i, \gamma)$
2	m=1 to M do:
3	$\tilde{y}_{im} = -\left[\frac{\partial \Psi(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}^{(x)}}$, i=1,…,N
4	$\{R_{lm}\}_1^J$ L-terminal node tree $(\{\tilde{y}_{im}, x_i\}_1^N)$
5	$\gamma_{lm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} \Psi(y_i, F_{m-1}(x_i) + \gamma)$
6	$F_m^{(x)} = F_{m-1}^{(x)} + v\gamma_{lm} I(x \in R_{lm})$
7	End

TABLE 2: Classification table of two methods for learning data.

	Learning Data		Predicted		Total
			CTS	Control	
AdaBoost	Observed	CTS	2100	214	2314
		Control	30	734	764
	Total		2130	948	3078
Gradient TreeBoost	Observed	CTS	2295	98	2393
		Control	92	765	857
	Total		2387	863	3250

APPLICATION

The individuals, who applied to Mersin University's Medical School's Neurology Main Scientific Branch's Electrophysiology Laboratory, with a prediagnosis of Carpal Tunnel Syndrome (CTS) were included in the study. The data set consists of 4076 incidences in total. 2520 (83.7%) of 3011 individuals with CTS taking place in the dataset were female patients and 491 (16.3%) of it were male patients. 869 (81.6%) of 1065 individuals in the control group were female and 196 (18.4%) of it were male. The conducted electrophysiological measurements are independent variables. The condition of being a patient with CTS or not was considered as the dependent variable. Boosting Tree application was conducted in Statistica 7.0 software package. Gradient TreeBoost application was conducted in Salford Predictive Modeler: TreeNet (R) trial version 6.6.0.091.

RESULTS

We compare the performance of combining tree model with that of two commonly used methods: AdaBoost and Gradient TreeBoost. Both models were randomly partitioned into a training set consisting of about one-third of the data, with the remaining data being used as the test set for demonstrating the performance of the algorithm. 3078 incidences of 4076 in total (2314 CTS + 764 healthy) were selected as learning data while the rest 998 incidences (697 CTS + 301 healthy) were selected as test data for AdaBoost algorithm application. 3250 incidences of 4076 in total (2393 CTS + 857 healthy) were selected as learning data while the rest 826 incidences (618CTS + 208 healthy) were selected as test data for application (Table 2).

In AdaBoost and Gradient TreeBoost algorithm, multiple trees are grown of the training data. 200 trees are produced for both models. 70 trees in the AdaBoost Algorithm (Figure 3) and 196 trees in the Gradient TreeBoost algorithm are chosen as the optimal trees (Figure 4). The error rates values of the trees formed belonging to the algorithms and the number of trees are given in Figure 3 and Figure 4.

Error rates as a function of the number of boosting with stumps for the AdaBoost procedure.

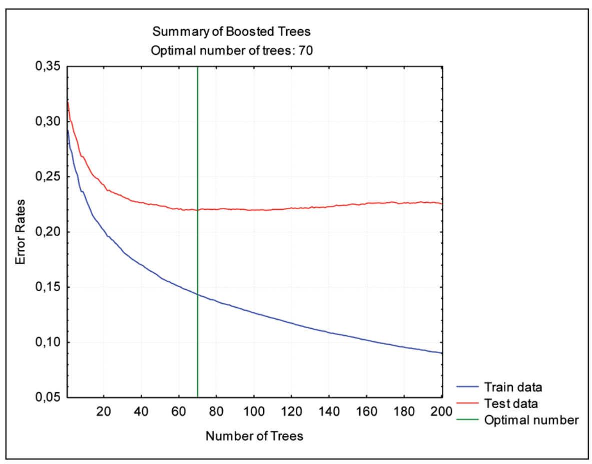

FIGURE 3: Error rates corresponding to the number of trees for AdaBoost Algoritm.

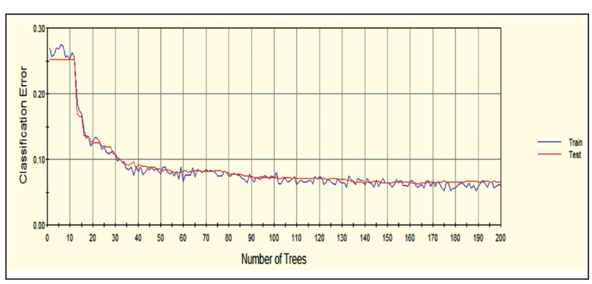

FIGURE 4: Error rates corresponding to the number of trees for Gradient TreeBoost Algoritm.

TABLE 3: Classification table of two methods for test data.

	Test Data		Predicted		Total
			CTS	Control	
AdaBoost	Observed	CTS	597	100	697
		Control	23	278	301
	Total		620	378	998
Gradient TreeBoost	Observed	CTS	586	32	618
		Control	24	184	208
	Total		610	216	826

TABLE 4: Classification successes for the learning and test data.

	AdaBoost			Gradient TreeBoost		
	Sensitivity (%) [Confidence Interval]	Specificity (%) [Confidence Interval]	Accuracy (%)	Sensitivity (%) [Confidence Interval]	Specificity (%) [Confidence Interval]	Accuracy (%)
Learning Data	90.75 [89.50-91.90]	96.07 [94.44-97.34]	92.07	95.90 [95.03- 96.66]	89.26 [87.00- 91.26]	**94**
Test Data	85.65 [82.83-88.17]	92.36 [88.75-95.09]	87.67	**94.82** [92.77- 96.43]	**88.46** [83.32-- 92.46]	**93**
P	0.0003	0.0078	<0.001	**0.2864**	**0.8351**	**0.3255**

Vertical line indicates the optimal iteration number of 70 (Figure 3).

Error rates as a function of the number of boosting with stumps for the Gradient TreeBoost procedure. Vertical line indicates the optimal iteration number of 196 (Figure 4).

The results of both method describe are shown in Table 2 and 3. In the tables, we reported classification table for learning data and test data.

In Table 4, the diagnostic test statistics of both procedures are reviewed. When the diagnostic test statistics are reviewed, the results of the learning data are higher in both procedures and these are the expected results (Table 4).

However, when the diagnostic accuracy statistics of the learning data and the test data are compared, the difference between the results of the Gradient TreeBoost Procedure is not statistically significant while the results for the learning data in the AdaBoost procedure are high ($p=0.2864$, $p=0.8351$, $p=0.3255$; Table 4). This situation might be considered as an indicator of a higher success in the accurate classification of new case by the Gradient TreeBoost procedure. Moreover, the results of the procedure are higher in every situation compared to the AdaBoost. Besides, the fact that the sensitivity or specificity values in the test data of the model are high indicates that they can be used as a successful method in CTS diagnosis.

DISCUSSION AND CONCLUSION

In medical applications, the choice of statistical methods for diagnosis of a given syndrome is an important topic. In parallel to developments on bioinformatics techniques, classification and regression methods based on decision trees have been frequently used for more reliable diagnosis.

Boosting based on combining tree has many advantages. Data sets can contain mixtures of nominal, ordinal and numerical variables. These methods are robust to outliers, missing data and correlated and irrelevant variables. These methods can also handle interactions, automatically select variables. It is believed that the boosting methods will become very more and more popular in health science due to its easy implementation and high predictive performance.

REFERENCES

1. Hastie TJ, Tibshirani R, Friedman J. Boosting and additive trees. The Elements of Statistical Learning. 2nd ed. New York: Springer; 2008. p.337-60.
2. Schapire RE. The strength of weak learns ability. Mach Learn 1990;5:197-227.
3. Friedman JH, Hastie T, Tibshirani R. Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). Ann Stat 2000;28(2):337-407.
4. Friedman JH. Greedy function approximation: A gradient boosting machine. Ann Stat 2001;29(5):1189-232.
5. Friedman JH. Stochastic gradient boosting. Comput Stat Data An 2002;38(4):367-78.
6. Freund Y, Iyer R, Schapire RE, Singer Y. An efficient boosting algorithm for combining preferences. J Mach Learn Res 2003;4:933-69.
7. Assaad M, Bon R, Cardot H. A new boosting algorithm for improved time-series forecasting with recurrent neural networks. Inform Fusion 2008;9(1):41-55.
8. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 1997;55(1):119-39.
9. Schapire RE, Singer Y. Improved boosting algorithms using confidence-rated predictions. Mach Learn 1999;37(3):297-336.
10. Ankaralı H, Orekici Temel G, Taşdelen B, Özge A. Boosting tree as a stronger approach in classification: An application of Carpal Tunnel syndrome. Journal of Turgut Ozal Medical Center 2012;19(4):228-33.
11. Borra S, Ciaccio A. Improving nonparametric regression methods by bagging and boosting. Comput Stat Data An 2002;38(4):407-20.
12. Ogutu LO, Piepho HP, Schulz-Streeck T. A comparison of random forests, boosting and support vector machines for genomic selection. BMC Proc 2011;5(Suppl 3):S11.
13. Cherkassky V, Mulier FM. Classification. Learning from Data: Concepts, Theory and Methods. 2nd ed. New Jersey: John Wiley&Sons; 2007. p.340-404.
14. Schapire RE, Freund Y, Bartlett P, Lee WS. Boosting the margin: a new explanation for the effectiveness of voting methods. Ann Stat 1998;26(5):1651-86.
15. Cao DS, Xu QS, Liang YZ, Zhang LX, Li MD. The Boosting: A new idea of building models. Chemometr Intell Lab 2010;100(1):1-11.
16. Friedman J, Meulman JJ. Multiple additive regression trees with application in epidemiology. Stat Med 2003;22(9):1365-81.
17. Zhang MH, Xu QS, Daeyaert F, Lewi PJ, Massart DL. Application of boosting to classification problems in chemometrics. Anal Chim Acta 2005;544(1-2):167-176.
18. Death G. Boosted trees for ecological modeling and prediction. Ecology 2007;88(1):243-51.
19. Breiman L, Friedman JH, Olshen RA, Stone CJ. Introduction to tree classification. Classification and Regression Trees. 1st ed. London: Chapman& Hall/CRC; 1984.p.18-55.
20. He P, Xu CJ, Liang YZ, Fang KT. Improving the classification accuracy in chemistry via boosting technique. Chemometrics and Intelligent Laboratory Systems 2004;70(1):39-46.
21. Rodriguez JJ, Maudes J. Boosting recombined weak classifiers. Pattern Recogn Lett 2008;29(8):1049-59.
22. Varmuza K, He P, Fank KT. Boosting applied to classification of mass spectral data. Journal of Data Science 2003;12(1):391-404.