Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 162 (2019) 946–954

Procedia Computer Science

www.elsevier.com/locate/procedia

7th International Conference on Information Technology and Quantitative Management (ITQM 2019)

Enhanced Classification Models for Iris Dataset

Yuanyuan Wu[a], Jing He*[a], Yimu Ji*[b], Guangli Huang[a], Haichang Yao[b], Peng Zhang[a], Wen Xu[a], Mengjiao Guo[a], Youtao Li*[c]

[a]*Swinburne University of Technology, Melbourne, Australia*
[b]*Nanjing University of Posts and Telecommunications, Nanjing, China*
[c]*Crystal's Wisdom Medical Technology Limited Company, Hefei, China*

Abstract

Data mining and machine learning are both useful tools in the field of data analysis. Classification algorithm is one of the most important techniques in data mining, therefore, it is of great significance to select suitable classification models with high efficiency to show superiority when solving classification problems with the use of Iris data. With this goal, a decision tree induction algorithm, namely graftedTree, is proposed to build randomized decision trees. Randomization is explicitly introduced into this algorithm, such that applying the algorithm several times on the same training data results in diversified models. An ensemble classification model is constructed using multiple randomized decision trees via majority voting. In order to show the performance of different models in classification, we propose the usage of precision, recall, F-Measure, the area under the ROC curve (AUC) and Gini coefficient as evaluation indexes of the classifying performance on the Iris dataset. The experimental results show that classification with Random Forests model has generally better performance than that with the Boosting Tree model and other three popular algorithms: KNN, SMO and Simple Cart. However, the Gini coefficient of the Random Forests model shows that it gets less pure training set than other models. The new GraftedTrees model inherits the advantages of Random Forest and further employs random mixture of two interchangeable node splitting rule inductions with the aim to obtain higher computational efficiency and better performance in terms of accuracy. With its superiority, it is expected that the new GraftedTrees model can prove to be the most powerful model with better performance in classification in the near future.

Peer-review under responsibility of the scientific committee of the 7th International Conference on Information Technology and Quantitative Management (ITQM 2019)

Keywords: GraftedTrees model; Random Forests model; Boosting Tree model; Iris dataset

1. Introduction

The classification problem has the longest history in the field of data mining, which is also a relatively thorough problem in research. Classification algorithm is one of the most important techniques in data mining. By analysing and comparing the latest and representative classification algorithms, the advantages and disadvantages of each type of algorithm are summarized, which is of great significance for researchers to choose the most suitable algorithm to deal with existing classification problems.

*Corresponding author.
E-mail address: jinghe@swin.edu.au, jiym@njupt.edu.cn, 45339573@qq.com.

1877-0509
Peer-review under responsibility of the scientific committee of the 7th International Conference on Information Technology and Quantitative Management (ITQM 2019)
10.1016/j.procs.2019.12.072

There are growing interest in "ensemble learning" - methods that generate many classifiers and aggregate their results [1]. Two well-known methods are Boosting and Bagging based on classification trees, and the Random Forests and the Boosting Tree are their representative algorithms respectively. The comparison results show that the traditional classification algorithm is not powerful enough. Therefore, it is of great importance to develop a more powerful algorithm of classification with better computational efficiency and better performance in accuracy than traditional classifying models.

As a variant of Random Forest, our proposed GraftedTrees algorithm is a new random decision tree based ensemble classifier, which is driven by the requirements of computational efficiency and good performance in terms of accuracy. It inherits the advantages of Random Forest: 1) within the ensemble, each individual tree can be trained parallelly. Within each individual tree, nodes from different branches can be trained independently; 2) fast prediction; 3) good performance. The difference between our method and Random Forest lies in the random decision tree induction algorithm. In our method, the GraftedTrees algorithm which employs random mixture of two interchangeable node splitting rule inductions: LDA-based multivariate splitting and random univariate splitting. It is anticipated that the new GraftedTrees model can find wider application in existing classification problems.

2. Classification Algorithm

As one of the best algorithms in classification [2], Random Forests algorithm is known as "the method representing the technical level of integrated learning". It is the representative algorithm based on Bagging developed on the basis of the decision tree [3]. The representative algorithm based on Boosting is called the Boosting Tree algorithm, which is also used with decision trees. As two widely used classification algorithms both based on the structure of decision trees, it is of great significance to compare their performances in classifying with the use of the Iris dataset.

The new GraftedTrees algorithm is a decision tree induction algorithm, which is proposed to build randomized decision trees. It is also a decision tree-based ensemble method with computational efficiency and good performance in terms of accuracy for classification problem. As a variant of Random Forest, it trains multiple random decision trees independently, and the ensemble combines these base learners using majority voting.

2.1. Boosting Tree Model

The Boosting Tree model [4] adopts the addition model (linear combination of basis functions), the forward step algorithm and the binary classification tree to deal with classification problems, which can also be regarded as the addition model based on decision trees. The Boosting Tree model generally has the following form:

$$f_M(x) = \sum_{m=1}^{M} T(x, \theta_m) \tag{1}$$

where T denotes the decision tree, and M is the number of trees. θ represents the parameter of the decision tree.

The forward step algorithm is used in the Boosting model. The original value for the algorithm is shown as follows:

$$f_0(x) = 0 \tag{2}$$

In mth step, the model can be written as:

$$f_m(x) = f_{m-1}(x) + T(x, \theta_m) \tag{3}$$

As for the value of the parameter θ of the decision tree, it can be determined by using the method of the empirical risk minimization:

$$\hat{\theta} = \arg\min_{\theta_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x) + T(x_i, \theta_m)) \tag{4}$$

2.2. Random Forests Model

Due to a large amount of computation in Neural Networks algorithm, regression tree and classification were brought up [7] in Random Forests, which reduce the processing time significantly by using binary method repeatedly. Then, according to the development method of random decision forests [8], the random forests are designed to be composed of classification trees [2], namely, the use of variables (columns) and data (rows) are randomized, which generates many classification trees. Then, the results of the classification trees are summarized as random forests, which significantly improves the accuracy of the Random Forests model.

The procedures are described as follows. In the kth tree, a random vector ϕ_k is formed, which is independent of the past random vectors $\phi_1, ..., \phi_{k-1}$ generated from previous steps but with the same distribution. Then, the tree grows with the use of training set and ϕ_k, producing a classifier $h(x, \phi_k)$, where x is an input vector. In random split selection, ϕ contains several independent random integers which are between the value of 1 and K. The value is determined by the use in the construction of tree for the nature and dimensionality of ϕ. After voluminous trees are produced, the most popular class is chosen by voting. These procedures generate the Random Forests model [2].

A random forest is a classifier composed of a collection of tree-structured classifiers $h(x, \phi_k), k = 1, ..., n$, where ϕ_k are random vectors which are independent identically distributed, and each tree casts a unit vote for the most popular class at input x.

There are classifiers $h_1(x)$, $h_2(x)$, ..., $h_k(x)$ as a group, and with the training set selected at random from the distribution of the random vector X, Y, the margin function is defined as follows.

$$mg(X, Y) = av_k I(h_k(X) = Y) - MAXav_k I(h_k(X) = j). \tag{5}$$

where I is the indicator function, and j is not equal to Y. The margin represents the extent to which the average votes at X, Y for the right class surpasses the average number of votes for any other wrong class. If the margin is larger, it means that the classification is more accurate. The generalization error is shown as follows:

$$PE^* = P_{X,Y}(mg(X, Y) < 0) \tag{6}$$

where the subscripts X and Y mean that the probability is over the X, Y space.

In the Random Forests model, $h_k(X) = h(X, \phi_k)$. For amounts of trees, they follow the Strong Law of Large Numbers. And as the number of trees increases, for almost surely all sequences $\phi_1, ..., \phi_n$, PE^* can be convergent to the following form [9]:

$$P_{X,Y}(P_\phi(h(X, \phi) = Y) - MAXP_\phi(h(X, \phi) = j) < 0) \tag{7}$$

where j is not equal to Y.

The process of random sampling in the Random Forests model ensures randomness, which avoids the phenomenon of overfitting. In the meanwhile, it also has many advantages compared to other models. Firstly, partial samples and features are selected for each tree, which contributes to avoiding overfitting to some extent. Secondly, it is random selections of samples and features for each tree that make the Random Forests model robust to noise attack and have stable performance. Thirdly, the Random Forests model has lower computational complexity and lower additional load. In the meanwhile, it also has the advantages of simple and easy implementation. Surprisingly, it shows powerful performance in various kinds of tasks. The Random Forests model only changes a little bit compared to Bagging, however, it is just the sample perturbation that results in the diversity of base learner in Bagging, which is different from that in the Random Forests model. The diversity of base learner in Random Forests model is not only from sample perturbation, but also from self-attribute perturbations, resulting in that the generalization ability in the final integration is further improved by increasing the differences between individual learners. Last but not least, the Random Forests model is able to handle high dimensional data without any operation of feature selection.

2.3. GraftedTrees Model

2.3.1. 1) LDA

is a linear classifier, which models $\frac{P(y=1|x)}{P(y=-1|x)}$ instead of modeling $P(y|x)$ directly. According to the Bayes theorem:

$$\frac{P(y=1|x)}{P(y=-1|x)} = \frac{p(x,y=1)}{p(x,y=-1)} = \frac{p(x|y=1)P(y=1)}{p(x|y=-1)P(y=-1)} \tag{8}$$

In LDAs modeling:

$$p(x|y=1) \sim N_1(u_1, \Sigma) \tag{9}$$

$$p(x|y=-1) \sim N_{-1}(u_{-1}, \Sigma) \tag{10}$$

$$p(y) \sim Born(\pi_1) \tag{11}$$

where N_1 and N_2 are two Gaussian distribution with respective mean and share a common covariance matrix, and $Born(\pi_1)$ is a Bernoulli distribution such that $P(y=1) = \pi_1$, $P(y=-1) = \pi_{-1} = 1 - \pi_1$. According to the Bayes decision rule under 0-1 loss, the decision boundary is given by:

$$\frac{p(y=1|x)}{p(y=-1|x)} = 1 \tag{12}$$

$$\Rightarrow \frac{p(x|y=1)P(y=1)}{p(x|y=-1)P(y=-1)} = 1 \tag{13}$$

$$\Rightarrow log\frac{p(x|y=1)P(y=1)}{p(x|y=-1)P(y=-1)} = 0 \tag{14}$$

$$\Rightarrow log\frac{N_1(x|u_1,\Sigma)\pi_1}{N_{-1}(x|u_{-1},\Sigma)\pi_{-1}} = 0 \tag{15}$$

Solving this equation, we can obtain:

$$\omega^T x + b = 0 \tag{16}$$

where

$$\omega = \Sigma^{-1}(u_1 - u_{-1}) \tag{17}$$

$$b = -\frac{1}{2}(\mu_1 + \mu_{-1})^T\Sigma^{-1}(u_1 - u_{-1}) + log\frac{\pi_1}{\pi_{-1}} \tag{18}$$

This is a linear decision boundary which is obtained analytically.

2.3.2. 2) GraftedTrees

The new GraftedTrees model can be viewed as a variant of Random Forest: multiple random decision trees are trained independently, and the ensemble combines these base learners using majority voting. We propose a random decision tree based ensemble classifier. The main idea of our design is as follows:

1) The optimization process of decision tree induction consists of two parts: the greedy top-down tree-growing and the node splitting at each node. At each splitting, decision induction follows the fundamental idea in statistical learning theory: 0-1 loss, Bayes‘ decision rule, and Bayes‘ risk. As a linear classifier, LDA follows the same fundamental idea as decision tree splitting rule. Hence, LDA can serve as splitting rule induction for decision tree.
2) For the convenience of optimization and control of complexity, most decision tree induction algorithms employ univariate splitting. The drawback is that univariate splitting can not capture the interaction between multiple input features. Instead, univariate splitting needs to be conducted multiple times to approximate such interaction. LDA is a multivariate linear classifier. It can capture the features interaction without incurring too much model complexity.
3) Decision tree has low bias error and high variance error. In another word, decision tree is at high risk of overfitting. There are three ways to reduce variance error: pre-pruning, post-pruning, and ensemble learning. The pre-pruning can be achieved via early-stopping implemented using various stopping criteria. The post-pruning normally is conducted via cross-validation, which is computational expensive. Ensemble learning can be employed to reduce variance error if the base learners are less dependent.

4) The key point in building a random decision forest as an ensemble is that: the base learner should be as accurate as possible and as diverse as possible. Accurate base learners can reduce the bias error while diverse learners can reduce the variance error.
5) There is no rigorous definition on what is intuitively perceived as diversity. In existing research work, the diversity of the base learner is introduced via different channels of randomness: random subsampling for the data instances, random selection of candidate splitting features, random selection of cutting points, etc. In our design, we explore a new channel of randomness: random mixture of LDA splitting and random univariate splitting.
6) The introduction of LDA as a multivariate linear splitting has two benefits: firstly, it can capture the interaction between features, which helps to reduce the depth of the tree; secondly, the randomness produced from the mixture of multivariate linear splitting and univariate splitting is the tree structure.
7) There is no rigorous quantification of randomness and analysis of randomnesss effect on ensemble classifiers performance. Hence, the control of randomness is important. In our design, we achieve the control of randomness from two aspects: a mixture parameter which controls the probability of using LDA splitting and random univariate splitting, and random selection of splitting features and cutting values in random univariate splitting rule induction.
8) LDA is a parametric method, and the training data become less as tree grows available. Hence, the estimation of parameters will be less reliable. Besides, empirical study shows that the interactions between features most likely arise at nodes close to the root. Therefore, in our design, we introduce a mixture shrinkage parameter which essentially gradually reduces usage of LDA splitting as tree depth grows.
9) Decision tree‘s partition of input space is hard-partition, which means a data sample will not be affected by splitting rules in another branch once it is partitioned into one branch. Ensemble of random decision trees can effectively mitigate this problem.

The pseudocode of our proposed method: Random forest of grafted decision trees (GraftedTrees) algorithm is as follows.

3. Experiments

We use the data of setosa and versicolor from Iris dataset as samples and apply Random Forests model and Boosting Tree model to do classification. Then, we take precision, recall, F-Measure, area under the ROC curve (AUC) and Gini coefficient as evaluation indexes, which can reflect the advantages of the two models in classification respectively during the comparison. Precision, recall, F-Measure, and AUC are the four most popular evaluation indexes when solving binary decision problems in machine learning. Precision is the proportion of correctly predicted samples to all positive samples, and recall is the ratio of predicted positive samples to all positive samples:

$$Precision = \frac{TP}{TP + FP} \tag{19}$$

$$Recall = \frac{TP}{TP + FN} \tag{20}$$

where TP is the number of samples that are tested positive as themselves. FP is the number of samples that are tested positive but negative and FN is the number of samples that are tested negative but positive. Hopefully, the higher both precision and recall are, the better the performance the model shows. But in fact, the two are at odds with each other in some cases. Therefore, F-Measure is proposed here, which stands for the harmonic mean of precision and recall:

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{21}$$

As another wildly used evaluation index of classification, AUC is the area under the ROC (Receiver Operating Characteristics) curve, which is also a popular evaluation index in the fields of data mining and machine learning. It can show the superiority of the classifier, especially faced with the problem of uneven distribution of datasets in different classes in comparison with the tradition indexes precision and recall. The ROC curve takes the false

Algorithm 1 Part 1

Input:

S, the training data set
X, the set of features
Y, the class label
N, the number of decision trees consisting the forest
θ, the mixture parameter of LDA_splitting and random_splitting criteria.
$\theta \in [0,1]$, default θ is set as 0.5
At each tree node to be split, LDA_splitting is selected at probability θ and random_splitting is selected at probability 1-θ.
λ, the regularization parameter for applying regularized LDA in LDA_splitting.
$\lambda \in [0,1]$, default λ is set as 0.1
Regularized LDA is adopted in LDA_splitting to solve singularity problem of pooled covariance matrix.
α, mixture shrinkage parameter
$\alpha \in [0,1]$, default α is set as 0.9. This parameter is used to reduce the probability of using LDA_splitting as the tree depth grows. For instance, given a node at depth t, the probability of using LDA_splitting is shrunk to $\theta\alpha^t$.
k, number of randomly selected candidate splitting features. This parameter is used in random_splitting. When random_splitting is applied, k features are randomly selected as candidate splitting features from the feature set X.
n_min, the minimum number of data samples for a node to be considered for splitting. During the decision tree learning process, when a node has less than *n_min* data samples, splitting will stop and that node will be marked as a leaf node.

Output: F, a random-forest-like ensemble of grafted decision trees

F=BuildForest($S, X, Y, N, \theta, \lambda, \alpha, k, n\_min$)
return F

function BuildForest($S, X, Y, N, \theta, \lambda, \alpha, k, n\_min$)
    $F = \emptyset$
    for $i = 1 \rightarrow N$ **do**
        $tree\_i = BuildTree(S, X, Y, N, \theta, \lambda, \alpha, k, n\_min)$
        $F = F + tree\_i$
    end for
    return F
end function

function BuildTree($S, X, Y, N, \theta, \lambda, \alpha, k, n\_min$)
    $T = R$
    $node\_splitting(R)$
    return T
end function

function node_splitting($node$)
    if $stop\_splitting(node) = TRUE$ **then**
        label node as a leafnode
    else
        $randomlypickanumberp \in [0, 1]$
        if $P < \theta$ **then**
            $(left\_child, right\_child) = LDA\_splitting(node)$
        else
            $(left\_child, right\_child) = random\_splitting(node)$
        end if
        node_splitting($left\_child$)
        node_splitting($right\_child$)
    end if
end function

Table 1: Values of evaluation indexes in two models

Model	Precision	Recall	F-Measure	AUC	Gini Coefficient
Random Forests	0.634	0.929	0.758	0.814	0.414
Boosting Tree	0.609	1	0.757	0.713	0.338
KNN	0.615	0.857	0.716	0.647	0.379
SMO	0.578	0.929	0.712	0.625	0.338
Simple Cart	0.614	0.964	0.75	0.670	0.291

positive rate (FPR) as the horizontal coordinate and the true positive rate (TPR) as the vertical coordinate, which are shown as follows [10]:

$$FPR = \frac{FP}{FP + TN} \tag{22}$$

$$TPR = \frac{TP}{TP + FN} \tag{23}$$

where, TN is the number of samples that are tested negative as themselves.

Gini coefficient is chosen as the other evaluation index to show the purity of data set, which further reflects the classification performance of the model. It is widely used in the CART algorithm, SPRINT algorithm, SLIQ algorithm, etc. The Gini coefficient is mainly described as follows [8]:

S is supposed as the set with s samples. The attributes of its class label have m different values respectively, which define m different classes $(C_i, i = 1, ..., m)$. S is divided into m subsets according to different values of class label attributes, namely, $(S_i, i = 1, ..., m)$. S_i is supposed to be the sample set C_i and s_i is the sample number of set S_i. Then, the Gini coefficient of S is:

$$Gini(S) = 1 - \sum_{i=1}^{m} P_i^2 \tag{24}$$

where P_i is the probability of any sample which belongs to C_i, and it can be estimated by s_i/s. When $Gini(S) = 0$ which means the instances in the set belong to the same class, it represents that the most useful information can be achieved; instead, when all samples are equally distributed, $Gini(S)$ can obtain its maximum value, which means just little useful information can be achieved. The basic idea of Gini coefficient is that every possible split method is traversed for each attribute, and when the minimum Gini coefficient is obtained, it will be chosen as the standard of node splitting regardless of root or child nodes, which is for the best performance of classification eventually.

In the experiment, the final training set with 100 data from two kinds of flower data in the Iris dataset is taken as the input for the two models, and the outputs are the values of evaluation indexes. As a comparison, we also choose KNN, SMO, and Simple Cart algorithms to show the advantages of the Random Forests model and the Boosting Tree model. The results are shown in Table 1.

From the results, we can see that the precision of the Boosting Tree model is 0.609, which is slightly lower than the value based on the Random Forests model. It is noticeable, however, that the Boosting Tree model has higher recall value than the one of the Random Forests model slightly, which reflects that they are nearly equally capable in recognizing positive samples from all positive samples. Generally, the Random Forests model performs slightly better than the Boosting Tree model according to the values of F-Measure. In the meanwhile, the AUC values of the Boosting Tree model and the Random Forests model are 0.717 and 0.814 respectively, which shows that the ensemble learning algorithms are of the great capability to do classification as well as their advantages in generalization ability. It is noticeable, however, the Gini coefficient of Boosting Tree model achieves 0.338, which outperforms the one of the Random Forests model with 0.414. It significantly shows that the Boosting Tree model can get more pure training set than the Random Forests model does but with lower AUC value as for the simulated astrophysical data.

Compared to the results of the other algorithms, the AUC of the Random Forests model is the highest among all the classification algorithms shown in the table with 0.814, which significantly indicates its superiority in classification. As for the values of F-Measure, the Random Forests model and the Boosting Tree model slightly

outperform other models, which are 0.758 and 0.757 respectively. From the aspect of the Gini coefficient, the other three models all perform better than the the Random Forests model, and the Simple Cart model has the purest training set from the results with the value of 0.291. Therefore, it is necessary to develop a more powerful classifying algorithm with better performance in classification in terms of classifying accuracy as well as efficient computation.

The new GraftedTrees algorithm inherits the advantages of Random Forest and further employs random mixture of two interchangeable node splitting rule inductions with the aim to obtain higher computational efficiency and better performance in terms of accuracy. It is anticipated that the new GraftedTrees model can have better performance in classification in the near future for its advantages.

4. Conclusion

This paper proposes to apply the Random Forests model and the Boosting Tree model to do classification on the Iris dataset. In this paper, precision, recall, F-Measure, AUC and Gini coefficient are taken as the evaluation indexes to measure the performance of the two models with the comparison to other three popular algorithms: KNN, SMO and Simple Cart, which can show their strong ability of generalization significantly. To be more exact, the results can significantly help researchers to choose the most suitable algorithm in dealing with existing classification problems. It is noticeable, however, the results of the Random Forests model are just slightly better than those on the basis of the Boosting Tree model. Thus, there can be chances to improve the performance with more enhanced classification model, making it possible to achieve higher accuracy of classification with more pure dataset.

In this paper, we propose a new decision tree-based ensemble method for classification problem, namely GraftedTrees algorithm, which inherits the advantages of Random Forest and further employs random mixture of two interchangeable node splitting rule inductions with the aim to obtain higher computational efficiency and better performance in terms of accuracy. With its superiority, it is expected that the new GraftedTrees model can prove to be the most powerful model with better performance in classification in the near future.

References

[1] Liaw, A., Wiener, M.: Classification and regression by randomForest. R news 2(3), 18-22 (2002)
[2] Roe, B. P., Yang, H. J., Zhu, J., Liu, Y., Stancu, I., McGregor, G.: Boosted decision trees as an alternative to artificial neural networks for particle identification. Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 543(2-3), 577-584 (2005)
[3] Iverson, L. R., Prasad, A. M., Matthews, S. N., Peters, M.: Estimating potential habitat for 134 eastern us tree species under six climate scenarios. Forest Ecology and Management 254(3), 390-406 (2008)
[4] Schapire, R.E.: The boosting approach to machine learning: An overview. Nonlinear estimation and classification pp. 149-171 (2003)
[5] Abney, S.: Bootstrapping. pp. 360-367. Proceedings of the 40th annual meeting of the Association for Computational Linguistics (2002)
[6] Breiman, L.: Bagging predictors. Machine Learning 24(2), 123-140 (1996)
[7] Breiman, L.: Random forests. Machine learning 45(1), 5-32 (2001)
[8] Stone, C.J.: Classification and regression trees. Wadsworth International Group (8), 452-456 (1984)
[9] Dong, L., Li, X., Xie, G.: Nonlinear methodologies for identifying seismic event and nuclear explosion using random forest, support vector machine, and naive bayes classification. Abstract and Applied Analysis (2014)
[10] Guo, C., Xu, J., Liu, L., Xu, S.: Maldetector-using permission combinations to evaluate malicious features of android app. pp. 157-160. The 6th IEEE International Conference on Software Engineering and Service Science (2015)

Algorithm 2 Part 2

function STOP_SPLITTING(*node*)
  if $|node| \leq n_min$ **then**
    return $TRUE$
  else
    if all samples at node have the same label **then**
      return $TRUE$
    else
      if all samples at node have the same feature values **then**
        return $TRUE$
      else
        return $FALSE$
      end if
    end if
  end if
end function

function LDA_SPLITTING(*node*)
  create two child nodes: left_child and right_child
  apply regularized LDA to the local data and learn a decision boundary: $w^T x + b = 0$
  $D^{(0)} = \{d_i = (x_i, y_i) | y_i = 0, d_i \in D\}$
  $D^{(1)} = \{d_i = (x_i, y_i) | y_i = 1, d_i \in D\}$
  $n = |D|$
  $n_0 = |D^{(0)}|$
  $n_1 = |D^{(1)}|$
  $\mu_0 = \frac{\sum_{d_i \in D^{(0)}} d_i}{n_0}$
  $\mu_1 = \frac{\sum_{d_i \in D^{(1)}} d_i}{n_1}$
  $\Sigma^{(0)} = \frac{1}{n_0 - 1}(\sum_{d_i \in D^{(0)}} (d_i - \mu^{(0)})^{(d_i - \mu^{(0)})^T})$
  $\Sigma^{(1)} = \frac{1}{n_1 - 1}(\sum_{d_i \in D^{(1)}} (d_i - \mu^{(1)})^{(d_i - \mu^{(1)})^T})$
  $\Sigma = \frac{(n_0 - 1)\Sigma^{(0)} + (n_1 - 1)\Sigma^{(1)}}{(n_0 - 1) + (n_1 - 1)}$
  $\hat{\Sigma} = \Sigma + \lambda I$
  $w = \hat{\Sigma}^{-1}(\mu^{(0)} - \mu^{(1)})$
  $b = -\frac{1}{2}(\mu^{(0)} + \mu^{(1)})^T \hat{\Sigma}^{-1}(\mu^{(0)} - \mu^{(1)}) + log\frac{n_0}{n_1}$
  split the local data at the current node into left_child and right_node accordingly
  left_child has the samples satisfy $w^T x + b < 0$
  right_child has the samples satisfy $w^T x + b \geq 0$
  return $left\_child, right\_child$
end function

function RANDOM_SPLITTING(*node*)
  create two child nodes: left_child and right_child
  randomly pick k candidate splitting features $\{a_i\}_{i=1}^k$ from the feature set X
  randomly draw a value v_i for each candidate splitting feature a_i
  calculate splitting rule: $(a^*, v^*) = argmaxScore(a_i, v_i)$, where Score is Gini index based
  split the local data at the current node into left_child and right_child accordingly
  left_child has the samples satisfying $x.a^* < v^*$
  right_child has the samples satisfying $x.a^* \geq v^*$
  return $left\_child, right\_child$
end function