Knowledge discovery from gene expression dataset using bagging lasso decision tree

Umu Sa'adah, Masithoh Yessi Rochayani, Ani Budi Astuti
Faculty of Mathematics and Natural Sciences, Universitas Brawijaya, Indonesia

Article Info

Article history:

Received Jun 18, 2020
Revised Aug 11, 2020
Accepted Sep 7, 2020

Keywords:

Bagging
Decision tree
Feature selection
Gene expression
High-dimensional

ABSTRACT

Classifying high-dimensional data are a challenging task in data mining. Gene expression data is a type of high-dimensional data that has thousands of features. The study was proposing a method to extract knowledge from high-dimensional gene expression data by selecting features and classifying. Lasso was used for selecting features and the classification and regression tree (CART) algorithm was used to construct the decision tree model. To examine the stability of the lasso decision tree, we performed bootstrap aggregating (Bagging) with 50 replications. The gene expression data used was an ovarian tumor dataset that has 1,545 observations, 10,935 gene features, and binary class. The findings of this research showed that the lasso decision tree could produce an interpretable model that theoretically correct and had an accuracy of 89.32%. Meanwhile, the model obtained from the majority vote gave an accuracy of 90.29% which showed an increase in accuracy of 1% from the single lasso decision tree model. The slightly increasing accuracy shows that the lasso decision tree classifier is stable.

Corresponding Author:

Umu Sa'adah
Department of Statistics
Universitas Brawijaya
Jalan Veteran, Malang, Indonesia
Email: u.saadah@ub.ac.id

1. INTRODUCTION

Gene expression data have been used to study the differences in gene characteristics between patients with certain diseases and normal people. The major challenge to analyze gene expression data is it has many predictors (genes), but the sample is much less. Gene expression data is a type of high-dimensional data that consist of thousands, even tens of thousands of gene features, but the sample size is only hundreds. Therefore, a certain strategy is needed to deal with dimensional problems in gene expression data.

One of the strategies in the classification of high-dimensional data is by reducing the dimension. There are two approaches in dimension reduction namely feature extraction and feature selection. The common dimension reduction approach in gene expression data is feature selection. Feature selection eliminates irrelevant and redundant features. Research [1] investigated the influence of feature selection on the accuracy of the classification of gene expression data. The result of the study was feature selection can increase accuracy up to 9%.

Several methods that combine feature selection and classification have been implemented in the classification of gene expression data. Assawamakin et.al. [2] used recursive feature elimination (RFE) to select genes and support vector machine (SVM) to classify several gene expression data. Kang et.al. [3] proposed a hybrid method of Relaxed Lasso and Generalized SVM for the multiclass classification of gene expression data. In the paper, Kang et.al. mentioned the selected genes, but these results are not validated based on theory. Gao et.al. [4] used information gain, gain ratio, relief, and correlation to select genes and SVM for classifying cancer. However, those studies only focused on the accuracy of the model obtained.

Accuracy, interpretability, and model size are three important aspects of machine learning [5], and many authors agree that model interpretability is the main concern [6]. In the classification of gene expression data, interpretation is very important besides the model accuracy. Classification and Regression Tree (CART), proposed by Breiman et.al. [7], is a Decision Tree algorithm that is promoted to be easily used in interpretation to explore knowledge from the data [8]. The CART algorithm is often used in decision analysis to visualize decision making. Besides that, the CART algorithm also can be used for feature selection. But CART requires high computational costs to work on very large data such as gene expression data. Therefore, applying the feature selection before applying CART is very useful.

Rochayani, et.al. [9] combined Lasso regularization and Decision Tree to select genes and classify gene expression data. The Lasso regularization has a low computational cost. Models obtained from the Lasso Decision Tree are also easy to interpret and theoretically correct. In this study, we are interested in examining the stability of the Lasso Decision Tree. A stable classifier is a classifier for which the prediction does not change much when the slight modification occurs in the training set.

Bagging, introduced by Breiman [10], is one of the ensemble methods that has been widely used to increase the accuracy of prediction models [11], improve the robustness and stability of the model [12, 13] and handle unbalanced class problems [14]. According to Breiman [10], improving the accuracy of Bagging depends on the stability of the classifier. Bagging usually could improve accuracy on unstable classifiers but cannot on stable classifiers. As an ensemble method, Bagging combines several single models into one final model based on the majority votes. Commonly, the single classifier for Bagging is the Decision Tree. But recently Bagging has been used in other machine learning algorithms such as Bagging Convolutional Neural Network [15] and Bagging Nearest Neighbor Support Vector Machine [16].

We performed the Bagging Lasso Decision Tree to examine the stability of the Lasso Decision Tree in modeling the gene expression dataset. The dataset used was the ovarian tumor dataset. It is interesting to use this dataset since ovarian cancer is one of the most fatal diseases in women which is frequently strikes post-menopausal women [17, 18]. The success of the two methods will be very useful in medical research, especially to discover new knowledge from a disease.

2. THEORETICAL BACKGROUND

2.1. Logistic regression

Binary logistic regression is used to model dataset with a binary response variable. The standard binomial logistic regression model is stated as a log of odds:

$$\log\left(\frac{\pi(x_i)}{1-\pi(x_i)}\right) = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j \tag{1}$$

where $\pi(x_i) = \frac{1}{1+e^{-\left(\beta_0+\sum_{j=1}^{p} x_{ij}\beta_j\right)}}$, $i = 1, 2, \ldots, n$ is the observation index, $j = 1, 2, \ldots, p$ is the index

of predictor variable, β_0 is the intercept, and β_j is the regression coefficient of j^{th} predictor variable. The estimation in logistic regression parameters was conducted by maximizing the log-likelihood function:

$$l(\beta_0, \beta_1, \ldots, \beta_p) = \frac{1}{n}\sum_{i=1}^{n}\left[y_i\left(\beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j\right) + \ln\left(1 + e^{\beta_0+\sum_{j=1}^{p} x_{ij}\beta_j}\right)\right] \tag{2}$$

Note that the first term of (2), $\beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j$, is a form of affine function, therefore it is concave. And the second term, $\ln\left(1 + e^{\beta_0+\sum_{j=1}^{p} x_{ij}\beta_j}\right)$, is also concave. Since the sum of concave functions is also concave, hence $l(\beta_0, \beta_1, \ldots, \beta_p)$ is a concave function and it implies that the negative log-likelihood, i.e. $-l(\beta_0, \beta_1, \ldots, \beta_p)$, is a convex function. The negative log-likelihood is also called the objective function of logistic regression. The advantage of convexity of negative log-likelihood function guarantees that the local optimum is also a global optimum. An optimization algorithm for a convex function such as the Newton method or the gradient descent can be implemented to solve the logistic regression problem. The estimated parameters of logistic regression are stated in (3).

$$\widehat{\boldsymbol{\beta}} = \arg\min\left[-l\left(\beta_0, \beta_1, \dots, \beta_p\right)\right]. \tag{3}$$

2.2. Regularization

Regularization is a method to avoid overfitting by adding constraints while solving the optimization problem. Let the regularization function is denoted by $R(\boldsymbol{\beta})$, where $\boldsymbol{\beta} = \left(\beta_1, \dots, \beta_p\right)$. In the regularized logistic regression, the constraint was added to the logistic regression loss function. The solution to the optimization problems stated in (4).

$$\widehat{\boldsymbol{\beta}}_R = \arg\min\left[-l\left(\beta_0, \beta_1, \dots, \beta_p\right)\right], \text{ such that } R(\boldsymbol{\beta}) \leq t. \tag{4}$$

The Lagrange form of the optimization problem in (4) is stated in (5).

$$\widehat{\boldsymbol{\beta}}_R = \arg\min\left[-l\left(\beta_0, \beta_1, \dots, \beta_p\right) + \lambda R(\boldsymbol{\beta})\right] \tag{5}$$

Where λ is a regularization parameter and is a positive real number. Consider the optimization problem in (5). When λ is set to zero, then the solutions $\widehat{\boldsymbol{\beta}}_R$ are the same as the solution of the unconstrained problem (full model) stated in (3). However, for high-dimensional problems (p>n), the λ cannot be set to zero because the saturated logistic regression fit is undefined [19].

Least Absolute Shrinkage Selection Operator (Lasso) is a popular regularization method introduced by Tibshirani [20]. Lasso works by adding the L1 penalty term, defined as $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p}\left|\beta_j\right|$, to shrink the coefficients of particular variables to be zero. Therefore, Lasso can be used for variable selection. The vector of estimated coefficients of the Lasso is stated as:

$$\begin{aligned} &\widehat{\boldsymbol{\beta}}_{lasso} \\ &= \arg\min\left[-l\left(\beta_0, \beta_1, \dots, \beta_p\right)\right. \\ &\left.+ \lambda\|\boldsymbol{\beta}\|_1\right] \end{aligned} \tag{6}$$

Since the coefficients of the regularization problem are controlled by the regularization parameter (λ), then the optimum λ should be estimated. K-fold cross-validation is used to estimate the optimum λ. The optimum λ is the one with the smallest average binomial deviance from the cross-validation results, i.e.

$$\hat{\lambda} = \underset{\lambda \in \{\lambda_1, \dots, \lambda_m\}}{\operatorname{argmin}} \operatorname{CV}(\lambda) \tag{7}$$

where

$$\operatorname{CV}(\lambda) = \frac{1}{n}\sum_{k=1}^{K} Dev_k(\lambda) \tag{8}$$

2.3. Decision tree

One of the Decision Tree algorithms is the Classification and Regression Tree (CART) that can be used when the predictor variables are categorical value or continuous value. The CART algorithm utilized the Gini impurity as the splitting criteria. The Gini impurity measure at node *t* was defined as:

$$i(t) = 1 - \sum_{j} p^2(c|t) \tag{9}$$

Where $i(t)$ was the Gini impurity and $p(c|t)$ was the proportion of class c in the node t. The Decision Tree construction begins with splitting binary the root node which contains all observations of the training set. The criteria used to determine the splitting of the root node is the goodness of split, denoted by $\Delta i(s,t)$, and defined as:

$$\begin{aligned} \Delta i(s,t) = i(t) - p_L[i(t_L)] \\ - p_R[i(t_R)] \end{aligned} \tag{10}$$

where:
s : a split,
p_L : proportion of observations at node *t* that go into the left child node t_L
p_R : proportion of observations at node *t* that go into the right child node t_R,
$i(t_L)$: impurity of the left child node, and
$i(t_R)$: impurity of the right child node.

To obtain the possible split-point for continuous-valued features, the observation values of the j$^{\text{th}}$ predictor, i.e. $\{x_{ij} | \, i = 1, \ldots, n\}$, are sorted in increasing order and ordered observation values $\{x_{(i)j}\}$ are obtained. Then, the midpoint of two adjacent values is considered as a possible split-point [21]. The point that had maximum goodness of split, i.e. $s = \max\{\Delta \, Gini(u_{cj})\}$ is selected as the split-point of the node [22]. The size of the tree is not limited. Therefore, splitting can be carried out to obtain more leaf nodes. However, large trees tend to overfit. The way to simplify the tree is to prune. The parameter to measure the complexity of a tree is called the complexity parameter (CP). To get the CP value, first, calculate resubstitution estimate and relative error. Resubstitution estimate is the proportion of misclassification in the training set [7]. Suppose T_a represents the subtree of the maximum tree T_{max} $(T_a < T_{max})$. Then the resubstitution estimate of T_a denoted by $R(T_a)$ is calculated using the formula in (11).

$$R(T_a) = \frac{1}{N}\sum_{i=1}^{N} I(T_a(x_i) \neq y_i) \tag{11}$$

Where, I is an indicator function that has value 1 if $T_a(x_i) \neq y_i$ and value 0 if $T_a(x_i) \neq y_i$, and N is the number of observations. Next, the relative error is obtained from the ratio of the resubstitution estimate of the subtree T_a and the resubstitution estimate at the root node or T_1. The relative error value is defined by (12).

$$Re(T_a) = \frac{R(T_a)}{R(T_1)} \tag{12}$$

where
$Re(T_a)$: relative error of the subtree T_a
$R(T_a)$: resubstitution estimate of the subtree T_a
$R(T_a)$: resubstitution estimate of the first subtree (subtree that only consists of a root node)
the complexity parameter is defined by (13).

$$CP_a = \frac{Re(T_a) - Re(T_{a+1})}{nsplit(T_{a+1}) - nsplit(T_a)} \tag{13}$$

where:
CP_a : the complexity parameters of subtree T_a
$Re(T_a)$: the relative error of subtree T_a
$nsplit(T_a)$: the number of the splitting of subtree T_a
A value of CP: 0 indicates no pruning which means the subtree is a maximum tree.

According to [7], to obtain the optimum Decision Tree, the one standard error rule (1 SE rule) is used. The 1 SE rule selects a model with a relative error of the cross-validation result ($CV(T_a)$) smaller or equal to $CV(T_a)$ minimum plus one standard deviation ($SD(T_a)$). The relative error of the cross-validation result is calculated using the formula in (14). Meanwhile, the standard deviation and standard error in the T_a subtree are calculated using (15) and (16).

$$CV(T_a) = \frac{1}{K}\sum_{k=1}^{K} Re({T_a}^{(k)}) \tag{14}$$

$$SD(T_a) = \sqrt{var\big(Re({T_a}^{(k)})) \ldots Re({T_a}^{(K)})\big)} \tag{15}$$

$$SE(T_a) = \frac{SD_k(T_a)}{\sqrt{K}} \tag{16}$$

the one standard error rule is presented in the (17).

$$\begin{aligned} &CV(T_a) \\ &\leq CV(\hat{T}_a) \\ &+ SE(\hat{T}_a) \end{aligned} \tag{17}$$

where

$$\hat{T}_a = \underset{T_a \in \{T_a,\dots,T_{max}\}}{\arg\min} CV_k(T_a) \tag{18}$$

2.4. Bootstrap aggregating (bagging)

The basic idea of Bagging is to use bootstrap resampling to get combined predictions. The first step of Bagging is bootstrap resampling that takes n samples from the training set with replacement to get the new training set. Furthermore, classification modeling is performed in B bootstrap replicates. The second step is aggregating, which is combining the estimation results into a single estimated value. The aggregating process is done by majority vote.

3. RESEARCH METHOD

We implemented the method in the OVA Ovary gene expression data available on http://openml.org. The dataset consists of 1,545 tumor tissue observations, in which 10,935 genes are observed on each of those tumor tissues. Observed tumor tissues are labeled with "Ovary" and "Other". The "Ovary" class is the class of ovarian tumor tissue; while the "Other" class is the class of other tumor tissues, including colon, breast, endometrial, kidney, lung, omental, prostate, and uterus tumor. The data analysis steps are described as follows.

3.1. Steps for predictor variables selection using lasso

a) Split the original dataset into the training and testing set. We used the ratio of 80% for the training set and 20% for the testing set.
b) Standardize the training set and run the Lasso regularization with 100 iterations.
c) Determine the optimum regularization parameter (λ) of the Lasso using 10-fold cross-validation
d) Extract the vector of coefficients ($\boldsymbol{\beta}_{Lasso}$) at the optimum λ

3.2. Steps for modeling a decision tree using the CART algorithm

a) Determine all possible split-point for each predictor variable.
b) Calculate the goodness of split $\Delta\, Gini(u_{cj})$ using the formula (7) to get the best split-point. The best split-point is $s = \max\{\Delta\, Gini(u_{cj})\}$.
c) Use the split-point s to split the root node binary so that the left child node and right child node are obtained.
d) Split the two child nodes until a maximum Decision Tree is formed.
e) Prune the maximum tree based on the rule stated in the (17) so that the maximum Decision Tree is obtained

3.3. Steps for bagging lasso decision tree

a) Perform sampling with replacement n times in the training set, where n states the size of observations in the training set.
b) Run the Lasso on the training set to select predictor variables.
c) Construct a Decision Tree using the CART algorithm on the training set that has been selected for predictor variables.
d) Repeat step 1 to step 3 B times to obtain B Decision Trees.
e) Combine B Decision Tree using majority votes.

The data analysis process was conducted using R. The R-packages used were glmnet for gene selection using Lasso regularization, rpart for generating the classification model, and rpart. plot for visualizing the classification model.

4. RESULTS AND DISCUSSION

The dataset was first split into the training set and the testing set with a ratio of 80%:20%. For the training set, 1,236 observations were obtained that would be used to construct the model. Meanwhile, for the testing set, 309 observations were obtained that would be used for evaluating the model. In the training set, standardization is performed, and then the Lasso regularization is run to get $\boldsymbol{\beta}_{Lasso}$. Because $\boldsymbol{\beta}_{Lasso}$ depends on the regularization parameter (λ), first, the optimum λ is estimated using 10-fold cross-validation. Figure 1 shows the result of 10-fold cross-validation to obtain the optimum λ of the Lasso.

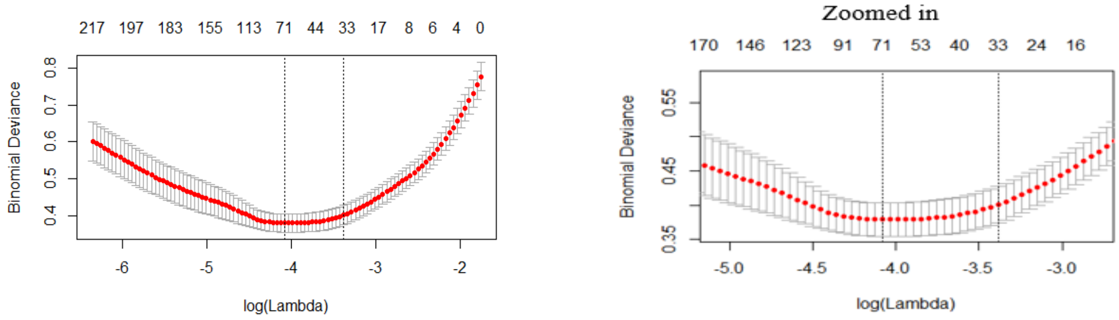
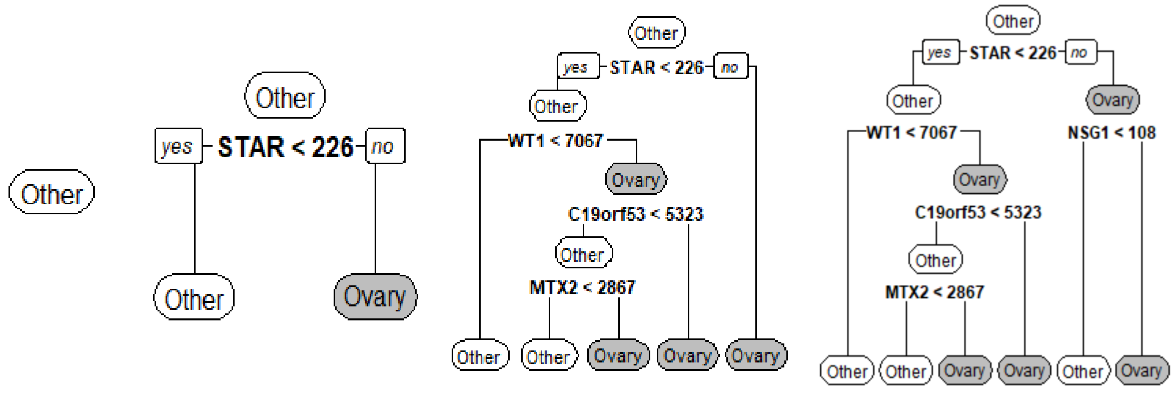


Figure 1. Estimating the optimum regularization parameter

The left-hand picture in Figure 1 shows the result of cross-validation and the right-hand zooms in the focused area. In both pictures, the vertical line on the left shows the minimum average binomial deviance, which is $\lambda_{\min} = 0.016917$. Meanwhile, the one on the right shows the one standard error of the minimum, which is $\lambda_{1\ \mathrm{SE}} = 0.0339902$. One can use the λ within one standard error of the minimum. For this case, we use $\lambda_{\min}$ as the optimum regularization parameter. The number along the top of the picture state the number of nonzero coefficients. Therefore, the optimum λ that we use produced 71 predictor variables with a nonzero coefficient. After the optimum λ is obtained, the coefficient $\hat{\beta}_j$ of the Lasso is then extracted.

4.1. Decision tree modeling

The steps to build a Decision Tree are to determine the Gini index from the training set, determine the node split-point, and determine the goodness of split. Splitting the nodes is done until getting the maximum tree and then prune it to get a simpler tree. Suppose that the maximum tree is denoted by T_{max} and the subtree of the maximum tree is denoted by T_a. To obtain the optimum tree, cross-validation is used. The optimum trees are subtrees that have $CV(T_a) \leq CV(\hat{T}_a) + SE(\hat{T}_a)$ as stated in (17). Figure 2 presents the subtrees of the maximum tree of the OVA_Ovary dataset with selected genes from Lasso. The result of cross-validation is displayed by the plot of the complexity parameter (CP) presented in Figure 3.



(a) T_1 (b) T_2 (c) T_3 (d) $T_4 = T_{max}$

Figure 2. Subtrees of the OVA_Ovary dataset with the selected genes from lasso (a) T_1, (b) T_2, (c) T_3, (d) $T_4 = T_{max}$

In Figure 3, the dotted horizontal line corresponds to the minimum cross-validation error plus one standard error of the minimum, $CV(\hat{T}_a) + SE(\hat{T}_a)$, which is 0.595. The optimum tree is indicated by a plot below the horizontal line. The number along the top of the plot states the number of leaf nodes produced by the subtree. Referring to Figure 3, the optimum subtrees are the subtree that produces 2, 5, and 6 leaf nodes because the plot is located below the horizontal line. Furthermore, from the three alternative optimum trees, we only use one to be the optimum. When the subtree with 2 leaf nodes is used, the model would perform poorly. And when the subtree with the 6 leaf nodes is used, the model would tend to overfit. Therefore, we use the subtree with 5 leaf nodes to be the optimum tree as shown in Figure 4.

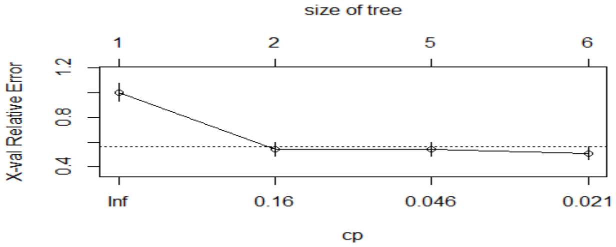


Figure 3. The plot of the complexity parameter

In Figure 4, the label on the node shows the class that has the largest proportion of that node. The compatibility of the model obtained from the optimum Decision Tree is measured using accuracy. The accuracy of the optimal Decision Tree is 95.31% for the training set and 89.34% for the testing set. This accuracy value is high even though the model only involves four predictors.

Besides evaluating the classification performance, we also interpret the model. Based on the model in Figure 4, to distinguish between ovarian tumor tissue and other tumors, first, the expression of the STAR gene is seen. If the STAR expression shows a value of more than or equal to 226, then the tumor tissue is predicted to be ovarian tumor tissue with a probability of 85%. Meanwhile, if the STAR expression value is less than 226, then the WT1 expression is seen. If WT1 expression is less than 7,067, it can be estimated that the tumor tissue is another tumor tissue with a probability of 97%. If WT1 shows an expression value greater than or equal to 7,067, then the C19orf53 gene expression is then seen. If the expression of C19orf53 is more than or equal to 5,323 then the tumor tissue must be ovarian, but if the C19orf53 expression is less than 5,323 then the expression of the MTX2 gene is also seen. If the MTX2 gene has more than or equal expression with 2,867, then it is predicted as ovarian tumor tissue, but if it is less than 2,867, then it is predicted as another tumor tissue.

The Decision Tree model obtained is theoretically correct. According to [23], STAR plays an important role in the production of steroid hormones. Estrogen is a steroid hormone that functions as a female sex hormone. Excessive estrogen can cause ovarian cancer. The theory supports the results of this study where ovarian tumor patients have higher STAR expression than other tumor patients. WT1 expression in ovarian tumors has been reviewed by [24, 25]. WT1 has a high expression in epithelial ovarian tumor patients (i.e. tumors in the ovary covering membrane) and the higher expression of WT1 indicates a high tumor stage [24]. Meanwhile, the results of research [25] stated that excessive WT1 expression can make cancer cells more aggressive in ovarian cancer. Meanwhile, the expression of C19orf53 and MTX2 genes in ovarian tumors has not been or has not been studied by researchers. However, C19orf53 and MTX2 are not important markers based on the model.

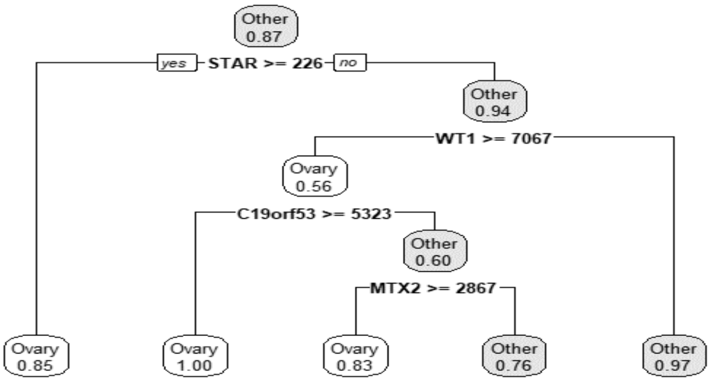


Figure 4. The optimum tree

4.2. Bagging lasso decision tree

We created $B = 50$ bootstrap replicates as done by [10] and in order not to require a long computation time. In the 50 bootstrap replicates, the Lasso was used to select the predictor variables, and then the Decision Tree model was constructed using the CART algorithm to obtain 50 single models. Of the 50 single models from the bootstrap replicates, a vote was carried out to combine the models into one final model. In the training set, the Bagging model gave an accuracy of 95.23%. Meanwhile, in the testing set, the Bagging model gave an accuracy of 90.29%. A summary of the accuracy values from the single model and the Bagging model is presented in Table 1.

Table 1. The accuracy of the single and the bagging

Model	Accuracy of the training set	Accuracy of the testing set
Single Lasso Decision Tree	95.31%	89.32%.
Bagging Lasso Decision Tree	95.23%	90.29%

Comparing the accuracy of the single models and the Bagging model has been conducted by [26]. In that study, a single Decision Tree model was compared with a Bagging Decision Tree with three splitting criteria, namely the gain ratio, information gain, and Gini index. The result is obtained that Bagging can increase accuracy by 2.96%, 1.65%, and 2.18%. In this study, an increase in accuracy is only 1% of the single Lasso Decision Tree model. This means that the Lasso Decision Tree classifier is stable on the OVA_Ovary dataset.

5. CONCLUSION

The Bagging Lasso Decision Tree was performed to examine the stability of the Lasso Decision Tree classifier. From the OVA Ovary dataset, the Lasso Decision Tree model was constructed with four splits but was able to produce an accuracy of 89.3% for the testing set. Meanwhile, the Bagging model gave an accuracy of 90.29%. The slightly increasing accuracy shows that the Lasso Decision Tree classifier is stable. The model obtained also gave a result that theoretically correct, beginning that the split-point of STAR was selected as a split-point of the root node. STAR was the important gene in estrogen hormone production. Therefore, researchers or practitioners concerning oncogenomic may use the Lasso Decision Tree method to study the difference in gene characteristics between normal and disease conditions and use Bagging Lasso Decision Tree to increase the accuracy.

ACKNOWLEDGEMENTS

We would like to express our deep gratitude to Prof. Widodo, S.Si., M.Si., Ph.D. Med.Sc., the head of Laboratorium Sentral Ilmu Hayati (LSIH) Universitas Brawijaya, for giving us knowledge about genes.

REFERENCES

[1] H. Omara, M. Lazaar, and Y. Tabii, “Effect of feature selection on gene expression datasets classification accuracy,” *Int. J. Electr. Comput. Eng.*, vol. 8, no. 5, pp. 3194-3203, 2018.

[2] A. Assawamakin, S. Prueksaaroon, S. Kulawonganunchai, P. J. Shaw, V. Varavithya, T. Ruangrajitpakorn, and S. Tongsima, “Biomarker selection and classification of “‘ - omics ’” data using a two-step bayes classification framework,” *Biomed Res. Int.*, 2013.

[3] C. Kang, Y. Huo, L. Xin, B. Tian, and B. Yu, “Feature selection and tumor classification for microarray data using relaxed lasso and generalized multi-class support vector machine,” *J. Theor. Biol.*, vol. 463, pp. 77-91, 2019.

[4] L. Gao, M. Ye, X. Lu, and D. Huang, “Hybrid method based on information gain and support vector machine for gene selection in cancer classification,” *Genomics. Proteomics Bioinformatics*, vol. 15, no. 6, pp. 389-395, 2017.

[5] A. Backhaus and U. Seiffert, “Neurocomputing Classification in high-dimensional spectral data : accuracy vs. interpretability vs . model size,” *Neurocomputing*, vol. 131, pp. 15-22, 2014.

[6] A. Bibal and B. Frénay, “Interpretability of machine learning models and representations : an introduction,” in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 77-82, 2016.

[7] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Chapman and Hall, 1984.

[8] A. Andrzejak, F. Langner, and S. Zabala, “Interpretable models from distributed data via merging of decision trees,” in *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pp. 1-9, 2013.

[9] M. Y. Rochayani, U. Sa’adah, and A. B. Astuti, “Two-stage gene selection and classification for a high-dimensional microarray data,” *J. Online Inform.*, vol. 5, no. 1, pp. 9-18, 2020.

[10] L. Breiman, “Bagging Predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123-140, 1996.

[11] H. R. Saad, “Industrial engineering & management use bagging algorithm to improve prediction accuracy for evaluation of worker performances at a production company,” *Ind Eng Manag.*, vol. 7, no. 257, pp. 2169-0316, 2018.

[12] P. Kim and K. Lim, “Vehicle type classification using bagging and convolutional neural network on multi view surveillance image,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 41-46, 2017.

[13] A. Ramaswamyreddy, S. Shiva, K. V Rangarao, and A. Saranya, “Efficient datamining model for prediction of chronic kidney disease using wrapper methods,” *Int. J. Informatics Commun. Technol.*, vol. 8, no. 2, pp. 63-70, 2019.

[14] N. W. S. Wardhani, M. Y. Rochayani, A. Iriany, A. D. Sulistyono, and P. Lestantyo, “Cross-validation metrics for evaluating classification performance on imbalanced data,” in *2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pp. 14-18, 2019.

[15] H. Li, Y. Li, F. Porikli, and M. Wang, “Convolutional neural net bagging for online visual tracking,” *Comput. Vis. Image Underst.*, vol. 153, pp. 120-129, 2016.

[16] I. Arieshanti, Y. Purwananto, A. Ramadhani, and M. U. Nuha, “Comparative study of bankruptcy prediction models,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 11, no. 3, pp. 591-596, 2013.

[17] P. Moorman, B. Calingaert, R. Palmieri, E. Iversen, R. Bentley, S. Halabi, A. Berchuck, and J. Schildkraut, “Hormonal risk factors for ovarian cancer in premenopausal and postmenopausal women,” *Am. J. Epidemiol.*, vol. 167, no. 9, pp. 1059-1069, Jun. 2008.

[18] F. Shen, S. Chen, Y. Gao, X. Dai, and Q. Chen, “The prevalence of malignant and borderline ovarian cancer in pre- and post-menopausal chinese women,” *Biomed Res. Int.*, vol. 8, no. 46, pp. 80589-80594, 2017.

[19] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall, 2015.

[20] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. R. Stat. Soc. Ser. B*, vol. 58, no. 1, pp. 267-288, 1996.

[21] J. Han, M. Kamber, and J. Pei, *Data Mining Concepts and Techniques Third Edition*. Elsevier, 2012.

[22] Y. Yohannes and P. Webb, “Classification and regression trees, cart - a user manual for identifying indicators of vulnerability to famine and chronic food insecurity. Washinton D.C,” *International Food Policy Research Institute*, 1999.

[23] P. R. Manna, C. L. Stetson, A. T. Slominski, and K. Pruitt, “Role of the steroidogenic acute regulatory protein in health and disease,” *Endocrine*, vol. 51, no. 1, pp. 7-21, 2016.

[24] B. Hylander, E. Repasky, P. Shrikant, M. Intengan, A. Beck, D. Driscoll, P. Singhal, S. Lele, and K. Odunsi, “Expression of Wilms Tumor gene (WT1) in epithelial ovarian cancer,” *Gynecol. Oncol.*, vol. 101, no. 1, pp. 12-17, 2006.

[25] Z. Liu, K. Yamanouchi, T. Ohtao, S. Matsumura, and M. Seino, “High Levels of Wilms’ Tumor 1 (WT1) expression were associated with aggressive clinical features in ovarian cancer,” *Anticancer Res.*, vol. 34, no. 5, pp. 2331-2340, 2014.

[26] A. Dzelihodzic and D. Donko, “Comparison of ensemble classification techniques and single classifiers comparison of ensemble classification techniques and single classifiers performance for customer credit assessment,” *Model. Artif. Intell.*, vol. 11, no. 3, pp. 140-150, 2016.