International Journal of Information Technology and Language Studies (IJITLS)

An Improvement in Branch and Bound Algorithm for Feature Selection

Ahood Naif Alharbi, Mohamed Dahab
aalharbi1482@stu.kau.edu.sa; mdahab@kau.edu.sa

Department of Computer Science, Faculty of Computing and Information Technology, King Abdul-Aziz University, Jeddah, Saudi Arabia

Abstract. Branch and bound (BB) algorithm undergoes an exponential growth in feature selection as the number of features increases, which may require, in the worst cases, exploring the whole tree looking for an optimal solution. This paper presents an enhancement in the BB algorithm for feature selection using an approximate monotonic criteria function. The enhanced version of the sub-optimal BB algorithm seeking for the solution by cutting unpromising paths and deleting multiple features at each internal tree node based on a predefined *tao* variable. The experiment was applied to different datasets and compared to the original BB algorithm and numerous selection methods. The results show promising results in terms of accuracy, elapsed time, and tree size.

Keywords: *Branch and Bound; Feature Selection; Dimensionality Reduction; Artificial Intelligent.*

1. Introduction

The central problem in machine learning is selecting a representative set of features used in constructing an easy and accurate classification model for a specific task. The problem known as dimensionality reduction, which is concerned about the space of hypothesis, indicates that an increase in the number of features leads to an exponential rise in hypothesis space (Alharbi and Dahab, 2018). The hypothesis is a function that predicts classes based on related data. The smaller the hypotheses space, the easier to find the right perdition hypotheses and vice versa. Such a process can be useful for both supervised and unsupervised learning problems.

Dimensionality reduction falls in one of the two known categories: Feature Selection or Feature Extraction. Feature extraction defined as the task where an initial set of raw variables (features) is reduced to more manageable groups for processing while still wholly and accurately describing the original data set (Kamel and Hadi, 2014; Cunningham, 2008).

While feature selection is the task that gets rid of redundant and irrelevant features to optimize the value of the criterion function J to reduce dimensionality (Liu and Motoda, 2007). In machine learning, three typical models are known as filter, wrapper, and embedded. The filter model is the fastest, easiest, and features are selected independently of the predicted model based on some criteria that rank the features previously and select the top m features. The wrapper model is more accurate but slower than the filter model because wrapper uses search algorithms to search through the space of possible features taking into account selecting features based on a prediction model and uses its performance to determine the quality of feature selection. The embedded model is selecting features during model building (Guyon et al., 2008; Sorzano et al., 2014). The following table below shows a comparison of the different characteristics of feature selection methods.

Selection Method	Search	Assessment	Advantages	Limitations	Examples
Filter	Rank features to specific criteria like information gain (individual feature ranking or nested subsets of features)	Use statistical tests	• Robust against over-fitting • Fast method • Simplicity	May fail to select the most useful features	Correlation, ANOVA, Information gain (Stein, 1975), etc.
Wrapper	Search the space of most useful feature subsets based on a specific machine learning algorithm	Use cross-validation	Can find the most valuable features	Prone to over-fitting Computationally expensive	Forward selection, Backward selection (Derksen, 1992), etc.
Embedded	Search guided by the learning process	Use cross-validation	Less computationally expensive than the wrapper	Prone to over-fitting	LASSO, Ridge Regression (Tibshira, 1996), etc.

Table 1. Feature Selection Methods.

The BB algorithm considers as a greedy search approach by evaluating all the possible combinations of features against the evaluation criterion and choose the best subset of features. The evaluation criterion is known as the performance measure, which depends on the given problem. For instance, the regression evaluation criterion could be p-values, R-squared, Adjusted R-squared, similarly for classification; the evaluation criterion could be accuracy, precision, recall, f1-score, etc. Ultimately, the BB algorithm selects the subset of features that gives the optimal solution for the specified machine learning algorithm.

The main objective of the study is to improve the original BB algorithm in feature selection in terms of consumed time and required resources. The findings should make an essential contribution to the field of machine learning that can potentially overcome the problem of dimensionality. Also, improving the BB algorithm is a practical step towards model interpret-ability.

This paper is divided into five parts. The first part is a literature review to show different algorithms performance in feature selection. The second part covers the original BB algorithm concept and definition with an example. The third part presents the new enhancement sub-optimal BB algorithm sincerely. The fourth part shows the experimental results, analysis, and discussion, while the last part is the conclusion and future work.

2. Literature Review

General dimensionality reduction techniques presented in the following Table (2).

Dimensionality Reduction Techniques	Definition
Percent Missing Values	Drop variables that have a high percentage of missing values. (No of records with missing values / No of total records) (Friedman et al., 1997).
Amount of Variation	Drop or review variables that have a very low variation (Lewontin and Hubby, 1966). $VAR(x) = \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$
Pairwise Correlations	If two variables are highly positive or negative correlated to each other, drop one of them will reduce dimensionality without loss of much information (Lewontin and Hubby, 1966).
Principal Component Analysis	Dimensionality reduction that emphasizes variation and uses orthogonal transformation (Malhi and Gao, 2004).

Correlation with the target	Drop variables that have a very low correlation with the target (Stein, 1975).
Forward Selection	It begins with an empty model and adds in variables one by one. In each forward step, add only variable that gives the single best improvement to the model (Derksen, 1992).
Backward Selection	In contrast to the forward selection, it begins with all the variables selected, and removes the least significant one at each step, until predefined criteria are satisfied (Derksen, 1992).
Stepwise Selection	Similar to the forward selection process, but a variable can also drop if it is deemed not useful anymore after a certain number of steps (Derksen, 1992).
Lasso (Least Absolute Shrinkage and Selection Operator)	Type of linear regression that uses shrinkage where data values shrunk towards a central point, like the mean (Tibshira, 1996).
Tree-Based	Fit several randomized decision trees on various sub-samples of the dataset and use averaging to rank order features (Guyon et al., 2008).

Table 2. Dimensionality Reduction Techniques.

In literature, many algorithms have been tested and evaluated to make the feature selection. Some algorithms guarantee to find the optimal subset of features m out of N features using specific criteria function J like exhaustive search and branch and bound (BB) algorithms (Jain and Zongker, 1997).

In contrast, other sub-optimal algorithms may miss the optimal solution like sequential forward selection (SFS) and sequential backward selection (SBS) (Pudil et al., 1994)(Marill and Green, 1963)(Whitney, 1971). Also, another research has used the genetic algorithm and HPC. They have accelerated the feature selection on the BCI dataset (the P300 based system). Genetic Algorithm (GA) and Differential Evolution (DE) are used as a search algorithm, while Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM) used as a classifier. DE-SVM has resulted in an accuracy of 80% selecting 42% of the original features only (Alwadei et al., 2017)(Alwadei et al., 2019).

Several papers worked in developing the BB algorithm in feature selection, precisely due to its exponential growth that increased as the number of features increased. In 1993, a paper used a more efficient BB+ algorithm in terms of reducing search time and elimination of some calculations by minimizing redundant J evaluations in the BB algorithm. BB+, like the original BB, finds the optimal subset of features by traversing the minimum solution tree starting from the root to the leaf nodes (Yu and Yuan, 1993).

Also, considerable effort invested into the acceleration of the BB algorithm via reducing the calculation of J function that usually computed in each internal node. The reduction process depends on predicting the value of the objective function based on the statistics of the effect of discarding individual measurements collected from previously evaluated feature sets (Somol et al., 2004). Another improvement to BB made in 2003, perform top-down and right-left search strategies with backtracking known as an improved branch and bound (IBB). They have utilized the information gained from the previous search and compared their results with BB, BB+, and FBB. IBB was faster than BB, BB+, and FBB (Chen, 2003).

Moreover, Casasent has suggested a new adaptive BB algorithm that has four characteristics. The tree constructed based on the importance of the features, and the large initial bound was set up using the floating search method. Also, Casasent has suggested a new approach that determines the level from which it starts looking to reduce J calculations (Nakariyakul and Casasent, 2007).

As seen in the literature, most of the enhancement papers struggle with the number of calculations and try to minimize it. However, the number of calculations of J criteria function noticeably increased as features increased, and the time, space complexity became unpractical in that situation. From this point of view, we have suggested an enhancement of the BB algorithm that constructs a practical solution tree by eliminating many features in each internal node.

3. The Branch and Bound algorithm

The branch and bound (BB) algorithm were developed early by Narendra and Fukunaga in 1977, describing the problem of selecting the optimal subset m features out of N whole set of features that met specific criteria function J (Narendra and Fukunaga, 1977). One required assumption in the BB algorithm is that J has to be a monotonic function, i.e.

$$m \in N \ then \ j(m) < j(N)$$

For a better demonstration of the algorithm, a visual example is shown in Figure (1). The tree has five features N and supposes m subset of features is equal two, and the J is a maximization criteria function that satisfies the monotonic condition.

$$J(x) = \sum_{\forall i \in x} i$$

Each node in the tree denotes a subset of features, and the root consists of N features. The number of tree levels determined by the following equation:

N-m+1

The number of expected combinations of m features computed by the following formula:

$$C_m^N = \frac{N!}{m! \ (N-m)!}$$

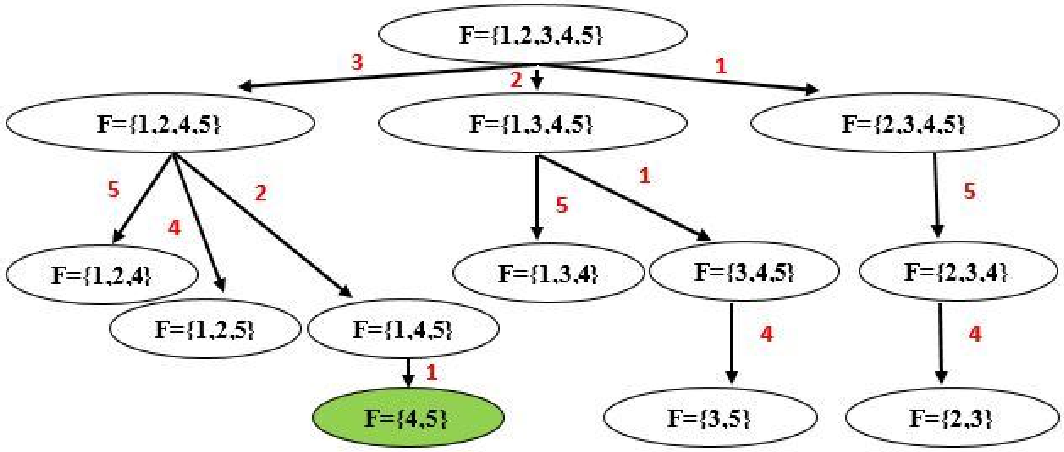

Figure 1. Branch and Bound Tree.

The further we go down the tree, one feature is eliminated at each level until the end up with the leaf nodes that have the only two candidate features m. The elimination process in the BB algorithm relies on selecting features that are not considered as preserved features. Preserved features for a particular node are those features that have deleted on its left side branches of the tree.

In the BB algorithm, traverse the tree from right to left in the depth-first search pattern. The bound B is set up in the rightmost value leaf node, and anytime the criterion value J(x) in some internal node is found to be lower than the current bound B, the whole subtree may be cut off, and many computations may discard due to the monotonic condition. Otherwise, the bound is updated, and the subtree is explored, promising a better solution (Nilsson, 2007). The powerful of the BB algorithm is the guarantee to find the optimal subset of features without examining the whole subtree due to the monotonic condition.

However, the BB algorithm is required for exploring all possible permutations in the worst-case, seeking the whole search space, and become an exhaustive search that costs 2^N calculations (Liu and Motoda, 2007). Also, as the number of features increases, the efficiency of the BB algorithm noticeably decrease in terms of time and the more computations it requires, i.e., the number of levels constructed when the number of features N is 20 and demand subset of features m is five equal to 16 levels. In the next section, we introduce a new enhanced sub-optimal BB feature selection algorithm, which handles the limitations of space and time.

4. The Proposed Enhancement Method of BB Algorithm

Both the original BB and the new enhanced BB algorithms perform 'top-down' search with backtracking. If N is a huge number like 5000, 200, and so on, we can not use the BB algorithm. Applying the concept of window (w) on a set of features will allow the BB algorithm to work practically on a large number of features' datasets. *Let*

$$F = \{ f_0, f_1, f_2, f_3, \dots, f_{N-1} \}$$

Where N is the number of all features, and f_i is the value of feature number i. The main goal is to maximize the objective function.

$$Z = \sum_{i=0}^{N-1} B_i X_i$$

Where Bi describes the benefit of feature i (the absolute value of the correlation of the feature and the output) and $X_i \in \{0,1\}$, $X_i=0$ means the feature is neglected while $X_i=1$ means the feature is used.

So, assuming N =20 features {f0,f 1,f 2 ,..,f 19} and w =5, which should be 4 ≤ w < 20 Then the selecting features would be as follows:

$$[f_i] = \{f_i, f_{i+w}, f_{i+2w}, \cdots\}$$

Where $[f_i]$ is the equivalent set of features. S is the selected features, S={f_1,f_2,f_3,f_4,f_5} where

$$S_i \cap S_j = \emptyset, \qquad S_i \cup S_j = S$$

Therefore, the sets will be equal to the window size as follows,

$$S_1 = [f_1] = \{f_0, f_5, f_{10}, f_{15}\}$$
$$S_2 = [f_2] = \{f_1, f_6, f_{11}, f_{16}\}$$
$$S_3 = [f_3] = \{f_2, f_7, f_{12}, f_{17}\}$$
$$S_4 = [f_4] = \{f_3, f_8, f_{13}, f_{18}\}$$
$$S_5 = [f_5] = \{f_4, f_9, f_{14}, f_{19}\}$$

The new version of the BB algorithm depends on eliminating more than one feature in each internal node, which reduces the number of constructed levels as well as accelerates the algorithm. The number of deleted features in each node is determined by the *tao* variable value, which best specified according to the correlation between the features.

Features correlation is a way to understand the relationship between multiple features in a given dataset. The relationship between the features could be decisive when both features A and B move in tandem, and they have a linear relationship or could be negative when feature A increases, then feature B decreases and vice versa. Features correlation can be determined easily by statistical methods like Spearman and Pearson Correlation Matrix (Adeli, 1999).

Each of those correlation types can exist in a spectrum represented by values from 0 to 1, where slightly positive correlation features can be between 0.5 and 0.7. while a strong relationship can be expressed by correlation score value of 0.9 or 1. If there is a strong negative correlation, it will be represented by an amount of -1. It is known that if the dataset has perfectly positive or negative correlated features, then there is a high chance that a problem named Multicollinearity will impact the performance of the model.

So, having prior knowledge about the correlation between a given set of features will allow us to best determine *tao* value in order to get rid of those correlated features.

J function in the enhancement BB algorithm is an Approximate Monotonic Branch and bound (AMBB) which allows non-monotonic functions to be used, typically classifiers, by relaxing the cut-off condition with the hope that these nodes will lead to a feasible solution rather than that terminates the search on a specific node.

A threshold has been defined based on the nature of the data, whether it is stationary (mean, variance, and autocorrelation structure) or not. The threshold will be bigger if the data are not stationary. In each internal node, the criteria function is computed, and the bound B is set up in the rightmost leaf. As the original BB algorithm, once the criterion value J(x) in some internal node was less than the current bound B, due to the approximate monotonic condition, the whole sub-tree may be cut off. Otherwise, the bound is updated, and the sub-tree is explored.

In figure(2), an example that used the new enhancement sub-optimal method consisting of 20-features artifact dataset, which contains 1000 samples. The target is 1 when (feature 1+ feature 2 < 7 or feature 4> feature 5 or feature 7%3==2) otherwise is 0. KNN (K-Nearest Neighbors) classifier was used as the J function that computes the highest accuracy of the subset of features m (Liao and Vemuri, 2002).

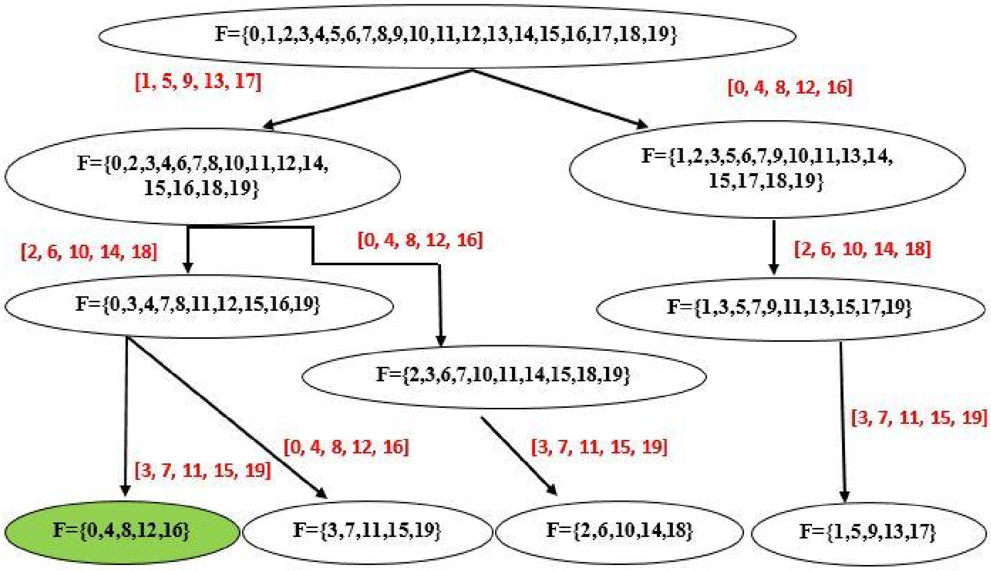
F={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}
[1, 5, 9, 13, 17]
[0, 4, 8, 12, 16]
F={0,2,3,4,6,7,8,10,11,12,14, 15,16,18,19}
F={1,2,3,5,6,7,9,10,11,13,14, 15,17,18,19}
[2, 6, 10, 14, 18]
[0, 4, 8, 12, 16]
[2, 6, 10, 14, 18]
F={0,3,4,7,8,11,12,15,16,19}
F={1,3,5,7,9,11,13,15,17,19}
F={2,3,6,7,10,11,14,15,18,19}
[3, 7, 11, 15, 19]
[0, 4, 8, 12, 16]
[3, 7, 11, 15, 19]
[3, 7, 11, 15, 19]
F={0,4,8,12,16}
F={3,7,11,15,19}
F={2,6,10,14,18}
F={1,5,9,13,17}

Figure 2. The Enhancement BB Algorithm Tree.

By exploring the correlation using Pearson correlation scores for all pairs of the 20 features in the dataset, the best *tao* to be used is 4. *tao* value used to determine how many features will be eliminated at each internal node; in this case, five features instead of 1 will be deleted in each internal node. The integer value, which determines the increment between each index for deletion, will depend on the length of the list of features in that node, for instance, at the root level, the deletion step will be four because 20 divided by 5 equals four and so on. The following table show step value for deletion at each level.

levels	No of Features	Step delation value
Level 0	20	4
Level 1	15	3
Level 2	10	2

Table 3. Deletion Step Value.

The number of levels decreased from 16 to 4 levels, and the candidate subset of features [0,4,8,12,16] scores 66.33% in 0.11 seconds elapsed time.

5. Experimental results and analysis

In this paper, the experiment was conducted on Fujitsu PRIMERGY CX400, Intel Xeon E5-2695v2 12C 2.4GHz, Intel TrueScale QDR. This machine called Aziz launched on June 01, 2015, at King Abdul-Aziz University (Fuj, 2015) and considered to be one of the top 500 supercomputers (Erich Strohmaier).

In the experiment, the KNN (K-Nearest Neighbors) classifier was used as the J function that computes the highest accuracy of the subset of features m. The performance has measured along with the change in the dataset features' number. The enhanced BB algorithm has tested across different real machine learning domain datasets called Wisconsin Diagnostic Breast Cancer (WDBC), Ionosphere Dataset, and Sonar Dataset obtained from UCI repository (William Wolberg, 1995)(Sejnowski,)(Sigillito, 1989).

The enhancement sub-optimal BB algorithm select *tao* based on the correlation between the features in each dataset. Pandas profiling is used to understand the relationship between multiple variables and attributes in the three datasets (McKinney, 2008).

The first part of the experiment is comparing the popular features selection methods with the new enhancement BB in terms of classification accuracy. The second part is mainly comparing the original BB algorithm with the new enhancement BB algorithm in terms of accuracy, elapsed time, and tree size.

5.1. The Enhanced BB Algorithm Versus Several Feature Selection Techniques

WDBC mammogram data is 2–class from the Wisconsin Diagnostic Breast Center (31 features, 357 benign, and 212 malignant samples). The Ionosphere Dataset requires the prediction of structure in the atmosphere given radar returns targeting free electrons in the ionosphere. It is a binary (2-class) classification problem. The number of observations for each class is not balanced. There are 351 observations with 34 input variables and one output variable.

Sonar Dataset involves the prediction of whether or not an object is a mine or a rock, given the strength of sonar returns at different angles. It is a binary (2-class) classification problem. The number of observations for each class is not balanced. There are 208 observations with 60 input variables and one output variable.

The desired number of features m can vary from the ones defined in table 4 according to selection field demand. In this experiment, we have defined m in Wisconsin Diagnostic Breast Cancer (WDBC) dataset as seven features out of 31, Ionosphere Dataset as seven features out of 34 and sonar dataset as four features out of 60).

In the Table (4), the enhancement sub-optimal BB algorithm often beat the other six known feature selection methods in classification accuracy by selecting the informative features in the three datasets. Although the BB algorithm has lost its optimality, it presents high accuracy in a short time. Such acceleration will be beneficial when the datasets have an enormous amount of features.

Selection Method	Dataset	No of Features (N)	Desired No of Features (m)	Classification Accuracy
The Enhancement BB Algorithm	Wisconsin Diagnostic Breast Cancer (WDBC)	31	7	91.97 %
	Ionosphere Dataset	34	7	87.72 %
	Sonar Dataset	60	4	65.24 %
PCA	Wisconsin Diagnostic Breast Cancer (WDBC)	31	7	73%
	Ionosphere Dataset	34	7	82%
	Sonar Dataset	60	4	62%
ANOVA	Wisconsin Diagnostic Breast Cancer (WDBC)	31	7	89.7%
	Ionosphere Dataset	34	7	68.7%
	Sonar Dataset	60	4	64.1%
Information Gain	Wisconsin Diagnostic Breast Cancer (WDBC)	31	7	89.7%
	Ionosphere Dataset	34	7	86.9%
	Sonar Dataset	60	4	65%
FCBF	Wisconsin Diagnostic Breast Cancer (WDBC)	31	7	88.1%
	Ionosphere Dataset	34	7	85.9%
	Sonar Dataset	60	4	63%

Gini Decrease	Wisconsin Diagnostic Breast Cancer (WDBC)	31	7	89.7%
	Ionosphere Dataset	34	7	85.9%
	Sonar Dataset	60	4	60.2%
X^2	Wisconsin Diagnostic Breast Cancer (WDBC)	31	7	89%
	Ionosphere Dataset	34	7	86.1%
	Sonar Dataset	60	4	64.2%

Table 4. The Enhancement BB Versus Several Feature Selection Techniques.

5.2. The Original BB Algorithm Versus the Enhanced BB Algorithm

The three datasets also have been tested against the original BB and the enhancement sub-optimal BB algorithm to compare the performance in terms of accuracy, elapsed time, and tree size. In Table (5), the enhanced BB algorithm has reflected efficient results in terms of accuracy, time, and tree size.

Dataset Name	No of Features (N)	No of Samples	Desired No of Features (m)	Algorithm	Classification Accuracy	Elapsed Time (seconds)	Tree Size (levels)
Wisconsin Diagnostic Breast Cancer (WDBC)	31	569	7	Original BB Algorithm	92.27 %	1.5	25
				Enhancement BB Algorithm	91.97 %	0.08	7
Ionosphere Dataset	34	351	7	Original BB Algorithm	86.32 %	1.33	28
				Enhancement BB Algorithm	87.72 %	0.29	10
Sonar Dataset	60	208	4	Original BB Algorithm	55.00 %	0.94	57
				Enhancement BB Algorithm	65.24 %	.09	15

Table 5. Different Datasets Performance Comparison.

The original BB algorithm will cost 2^N possible feature subsets in the worst case (where N is the number of features), which is computationally impractical. Also, tree size is a very influential factor, especially if the number of features is extensive. WDBC has only two leaf nodes, and seven levels constructed in .08 seconds and resulted in 91.97 % using the enhanced BB algorithm. While the original BB needs 25 levels and 16963900000 leaf nodes to find the optimal solution.

Despite that, the accuracy in WDBC using the original BB algorithm is higher than the enhancement sub-optimal BB algorithm, which is expected, but on the two other datasets, the enhancement BB algorithm has proven its strength by achieving better accuracy.

As seen in figure (4), the time is noticeably has improved and reflected better accuracy than the traditional BB version. The consumed amount of time in calculations will increase if the number of features increases.

In Figure (5), we have shown how bigger the tree in both algorithms by counting the number of levels in the different number of features data sets. The tree is being enormous when features are eliminated one by one in each internal node, and the tree file cannot be opened. On the other hand, the enhanced BB algorithm constructs a more practical and reasonable tree in minimum time and higher accuracy.

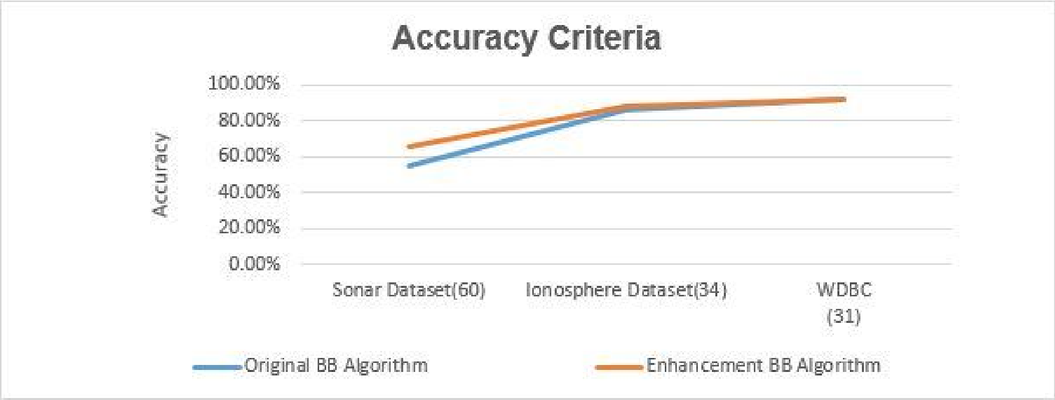

Figure 3. Accuracy Criteria.

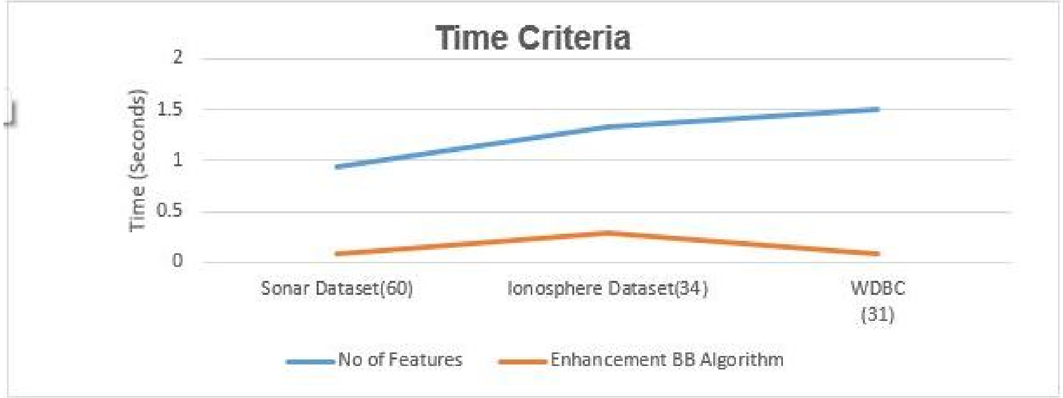

Figure 4. Time Criteria.

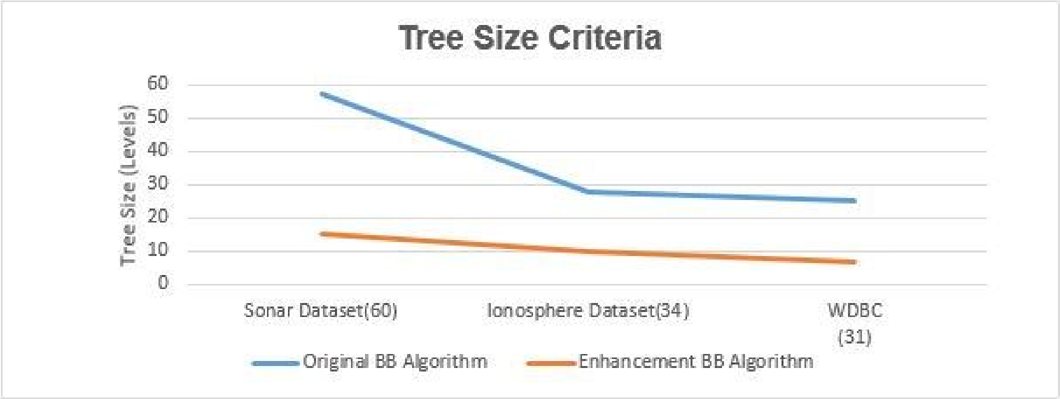

Figure 5. Tree Size Criteria.

6. Conclusion and future work

Based on a detailed study on the Branch and Bound algorithm concept in feature selection, we have enhanced the algorithm noticeably in terms of the number of computations and tree size. The improved BB algorithm relies on eliminating more than one feature in each internal node based on features' correlation. We have tested the enhanced BB across the different number of features datasets taking into account accuracy, elapsed time, and tree size criteria. The improved BB algorithm has been tested against different selection methods. The results have shown an efficient performance compared to the original BB traditional algorithm and the other feature selection techniques. As future work, we will use more than

10,000 features data sets and parallel the enhanced algorithm implementing using high-performance computing.

References

Adeli, H. (1999). Machine Learning-Neural Networks, Genetic Algorithms, and Fuzzy Systems. Kybernetes.

Alharbi, A. N., & Dahab, M. (2018). Comparative Study on Fast Feature Selection. International Journal of Information Technology and Language Studies, 2(2).

Alwadei, M. D. S., Dahab, M., & Kamel, M. (2017). A Feature Selection Model based on High-Performance Computing (HPC) Techniques. International Journal of Computer Applications, 180(7), 11-16.

Alwadei, S., Dahab, M., & Kamel, M. (2019). High performance GA-LDA feature selection model for Brain-Computer Interface data. International Journal of Information Technology, 3(1), 1-12.

Chen, X. W. (2003). An improved branch and bound algorithm for feature selection. Pattern Recognition Letters, 24(12), 1925-1933.

Cunningham, P. (2008). Dimension reduction Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval.

Derksen, S., & Keselman, H. J. (1992). Backward, forward, and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. British Journal of Mathematical and Statistical Psychology, 45(2), 265-282.

Erich Strohmaier. Erich Strohmaier, Jack Dongarra, H. S. M. M. top 500. https://www.top500.org/site/50585. Accessed: 2019-11-4.

Friedman, N. (1997, July). Learning belief networks in the presence of missing values and hidden variables. In ICML (Vol. 97, No. July, pp. 125-133).

Fuj. (2015). Fujitsu Supports King Abdul-Aziz University Research Capabilities with New Supercomputing System. King Abdul-Aziz University. Fujitsu Limited.

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (Eds.). (2008). Feature extraction: foundations and applications (Vol. 207). Springer.

Hubby, J. L., & Lewontin, R. C. (1966). A molecular approach to the study of genic heterozygosity in natural populations. I. The number of alleles at different loci in Drosophila pseudoobscura. Genetics, 54(2), 577.

Jain, A., & Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. IEEE transactions on pattern analysis and machine intelligence, 19(2), 153-158.

Kamel, M. I., & Hadi, A. A. (2014). Improving P300 based speller by feature selection. Journal of Medical Imaging and Health Informatics, 4(4), 469-487.

Liao, Y., & Vemuri, V. R. (2002). Use of k-nearest neighbor classifier for intrusion detection. Computers & Security, 21(5), 439-448.

Liu, H., & Motoda, H. (Eds.). (2007). Computational methods of feature selection. CRC Press.

Malhi, A., & Gao, R. X. (2004). PCA-based feature selection scheme for machine defect classification. IEEE Transactions on Instrumentation and Measurement, 53(6), 1517-1525.

Marill, T., & Green, D. (1963). On the effectiveness of receptors in recognition systems. IEEE transactions on Information Theory, 9(1), 11-17.

McKinney, W. (2008). python data analysis library. http://pandas.sourceforge.net. Accessed: 2019-10-4.

Nakariyakul, S., & Casasent, D. P. (2007). Adaptive branch and bound algorithm for selecting optimal features. Pattern Recognition Letters, 28(12), 1415-1427.

Narendra, P. M., & Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. IEEE Transactions on Computers, (9), 917-922.

Nilsson, R. (2007). Statistical feature selection: with applications in life science (Doctoral dissertation, Institutionen för fysik, Kemi och biologi).

Pudil, P., Ferri, F. J., Novovicova, J., & Kittler, J. (1994, October). Floating search methods for feature selection with nonmonotonic criterion functions. In Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5) (Vol. 2, pp. 279-283). IEEE.

Schneidman, E., Berry II, M. J., Segev, R., & Bialek, W. (2006). Weak pairwise correlations imply strongly correlated network states in a neural population. Nature, 440(7087), 1007.

Sejnowski, T. Sonar dataset from UCI. https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks). Accessed: 2019-11-4.

Sigillito, V. (1989). Ionosphere dataset from UCI. https://archive.ics.uci.edu/ml/datasets/Ionosphere. Accessed: 2019-10-15.

Somol, P., Pudil, P., & Kittler, J. (2004). Fast branch & bound algorithms for optimal feature selection. IEEE Transactions on pattern analysis and machine intelligence, 26(7), 900-912.

Sorzano, C. O. S., Vargas, J., & Montano, A. P. (2014). A survey of dimensionality reduction techniques. arXiv preprint arXiv:1403.2877.

Stein, J. J., & Blackman, S. S. (1975). Generalized correlation of multi-target track data. IEEE Transactions on Aerospace and Electronic Systems, (6), 1207-1217.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.

Whitney, A. W. (1971). A direct method of nonparametric measurement selection. IEEE Transactions on Computers, 100(9), 1100-1103.

William Wolberg. (1995). William Wolberg, Nick Street, O. M. (1995). Wisconsin diagnostic breast cancer (wdbc) dataset from UCI. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic). Accessed: 2019-10-4.

Yu, B., & Yuan, B. (1993). A more efficient branch and bound algorithm for feature selection. Pattern Recognition, 26(6), 883-889.