Journal of Scientific Research & Reports
9(1): 1-9, 2016; Article no.JSRR.19523
ISSN: 2320-0227

SCIENCEDOMAIN *international*
www.sciencedomain.org

Decision Tree Classifiers Based on Granular Computing

Hongbing Liu[1*], Fan Zhang[1] and Chang-An Wu[1]

[1]School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China.

Authors' contributions

This work was carried out in collaboration between all authors. Author HL designed the study, wrote the protocol, and wrote the first draft of the manuscript. Author FZ managed the literature searches, analyses of the study performed the spectroscopy analysis and author CAW managed the experimental process and results. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JSRR/2016/19523
Editor(s):
(1) Saad Mohamed Saad Darwish, Department of Information Technology, Institute of Graduate Studies and Research (IGSR), University of Alexandria, Egypt.
(2) Fabrício Moraes de Almeida, Federal University of Rondônia, Porto Velho, RO – Brasil.
(3) Martin J. Bohner, Missouri University of Science and Technology, Rolla, Missouri, USA.
(4) Narayan Thapa, Department of Mathematics and Computer Science, Minot State University, 58707 Minot, ND, USA.
Reviewers:
(1) Anonymous, University of Port Harcourt, Port Harcourt, Nigeria.
(2) Anonymous, Middle Technical University, Iraq.
(3) Anonymous, G Pulla Reddy College of Engineering and Technology, India.
Complete Peer review History: http://sciencedomain.org/review-history/11508

Original Research Article

Received 14th June 2015
Accepted 4th September 2015
Published 23rd September 2015

ABSTRACT

Bottle-up and top-down are two main computing models in granular computing (GrC). The bottle-up granular computing is used to form decision tree classifiers, or DTCGrC for short. Algorithm DTCGrC constructs a framework of granular computing by the bottle-up join operation which maps all the training data into the granule set, and the achieved granule set is used to form the decision tree classifiers. We compare the performance of DTCGrC with decision tree classifiers, for a number of two-class problems and multiclass problems. Our computational experiments showed that DTCGrC improves the generalization abilities.

Keywords: Decision tree classifier; granular computing; hypersphere granule.

**Corresponding author: Email: liuhbing@126.com;*

1. INTRODUCTION

Decision tree learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. It is one of the predictive modelling approaches used in statistics, data mining and machine learning [1-3]. Tree models where the target variable can take a finite set of values are called classification trees [4]. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees [5].

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data but not decisions; rather the resulting classification tree can be an input for decision making.

Granular computing (GrC) concerns the processing of complex information entities called information granules, which arise in the process of data abstraction and derivation of knowledge from information or data [6,7]. In the philosophical sense, granular computing can describe a way of thinking that relies on the human ability to recognize the real world under various levels of granularity in order to abstract and consider only those things that serve a specific interest and to switch among different granularities. By focusing on different levels of granularity, one can obtain different levels of knowledge, as well as a greater understanding of the inherent knowledge structure. Granular computing is thus essential in human problem solving and hence has a very significant impact on the design and implementation of intelligent systems, such as classification problems [8-13].

In this paper, we present the decision tree classifiers based on granular computing. Firstly, the granule is represented as the form of hypersphere. Secondly, the operation $\vee$ is introduced to transform the granule space with the smaller granularity into the granule space with the larger granularity. Thirdly, the threshold of granularity is used to control the operation between two granules. Finally, the granule set induced by operation $\vee$ is used to form the decision tree classifiers.

2. A BOTTLE-UP FRAMEWORK OF GrC

For the data set $S=\{\boldsymbol{x}_i|i=1,2,...,n\}$ in N-dimensional space, GrC algorithm is formed in terms of the following steps. Firstly, the representation method of granule is proposed. Secondly, operations between two granules are designed. Thirdly, the fuzzy inclusion relation between two granules is measured by fuzzy inclusion measure. Finally, the GrC algorithms are designed by operations between two granules.

2.1 Representation of Granules

A granule is represented as a subset of S which is composed by the data with the similar features, and the size of granule is measured by the granularity induced by the maximal distance between data belonging to the same granule. In order to facilitate the study of granular computing, such as the operations between two granules, the granules are represented as the standard form, for example, the granule with the shape of circle in 2-dimensional space and the granule with the shape of hypersphere N-dimensional space.

A granule is represented as the hypersphere $\boldsymbol{G}=(\boldsymbol{C},R)$, where $\boldsymbol{C}$ is the center of granule, R is radii of granule, and refers to the granularity of granule $\boldsymbol{G}$ which is measured by the maximal distance between center and the data included in granule. Particularly, a point $\boldsymbol{x}$ is represented by a atomic granule with the center $\boldsymbol{x}$ and granularity 0 in N-dimensional space. The distance between center $\boldsymbol{C}=(c_1,c_2,...,c_N)$ and datum $\boldsymbol{x}=(x_1,x_2,...,x_N)$ can be defined as follows:

$$d_1(x,C)=((x_1-c_1)^2+(x_2-c_2)^2+...+(x_N-c_N)^2)^{1/2}$$

2.2 Operations Between two Granules

The operations between two granules reflect the transformation between macroscopic and microcosmic of human cognitions. When a person want to observe the object more carefully, the object is partitioned into some suitable sub-objects, namely the universe is transformed into some parts in order to study the object in detail in the view of microscopic. Conversely, there is the same attributes of some objects, we regard the objects as a universe to simple the process in the view of macroscopic. The operations between two granules are designed to realize the transformation between macroscopic and microscopic. Set-based models of granular structures are special cases of lattice-based models, where the lattice join operation $\vee$ coincides with set union operation $\cup$ and lattice meet operation $\wedge$ coincides with set intersection operation $\cap$. Join operation $\vee$ and meet operation $\wedge$ are used to realize the transformation between macroscopic and microcosmic. Operation $\vee$ unites the granules with small granularities to the granules with the large granularities. Inversely, Operation $\wedge$ divides the granules with large granularities into the granules with small granularities. Join operation $\vee$ and meet operation $\wedge$ are designed as follows.

Any points are regarded as atomic granules which are indivisible, the join process is the key to obtain the larger granules compared with atomic granules. Likewise, the whole space is a granule with the maximal granularity, the meet process produces the smaller granules compared with original granules.

For two hypersphere granules $\boldsymbol{G}_1=(\boldsymbol{C}_1, R_1)$ and $\boldsymbol{G}_2=(\boldsymbol{C}_2, R_2)$ in N-dimensional space, the join hypersphere granule is

$$\boldsymbol{G}=\boldsymbol{G}_1\vee\boldsymbol{G}_2=(\boldsymbol{C}, R)$$

The center $\boldsymbol{C}$ and the granularity R of $\boldsymbol{G}$ are computed as follows.

Firstly, the vector from $\boldsymbol{C}_1$ to $\boldsymbol{C}_2$ and vector from $\boldsymbol{C}_2$ to $\boldsymbol{C}_1$ are computed. If $\boldsymbol{C}_1=\boldsymbol{C}_2$, then $\boldsymbol{C}_{12}=0$ and $\boldsymbol{C}_{21}=0$. If $\boldsymbol{C}_1\neq\boldsymbol{C}_2$, then $\boldsymbol{C}_{12}=(\boldsymbol{C}_2-\boldsymbol{C}_1)/d(\boldsymbol{C}_1,\boldsymbol{C}_2)$ and $\boldsymbol{C}_{21}=(\boldsymbol{C}_1-\boldsymbol{C}_2)/d(\boldsymbol{C}_2,\boldsymbol{C}_1)$.

Secondly, the crosspoints of $\boldsymbol{G}$ and $\boldsymbol{G}_1$ are $\boldsymbol{P}_1=\boldsymbol{C}_1-\boldsymbol{C}_{12}R_1$ and $\boldsymbol{P}_2 = \boldsymbol{C}_1+\boldsymbol{C}_{12}R_1$. The crosspoints of $\boldsymbol{G}$ and $\boldsymbol{G}_2$ are $\boldsymbol{Q}_1 = \boldsymbol{C}_2-R_2\boldsymbol{C}_{21}$ and $\boldsymbol{Q}_2 = \boldsymbol{C}_2+R_2\boldsymbol{C}_{21}$.

Thirdly, the center C and granularity R of the join hypersphere granule $\boldsymbol{G}$ is computed by algorithm 1.

Algorithm 1. Computing C and R of join hypersphere granule G between G_1 and G_2

```
Input: $G_1=(C_1,R_1)$ and $G_2=(C_2,R_2)$
Output: $G=(C,R)$
if $R_1>=R_2$
    if d$(C_1,C_2)<=R_1-R_2$   $C=C_1$             $R=R_1$
    else                       $C=(P_1+Q_1)/2$     $R=d(P_1,Q_1)/2$
    end
else
    if $d(C_1,C_2)<=R_2-R_1$   $C=C_2$             $R=R_2$
    else                       $C=(P_1+Q_1)/2$     $R=d(P_1,Q_1)/2$
    end
end
```

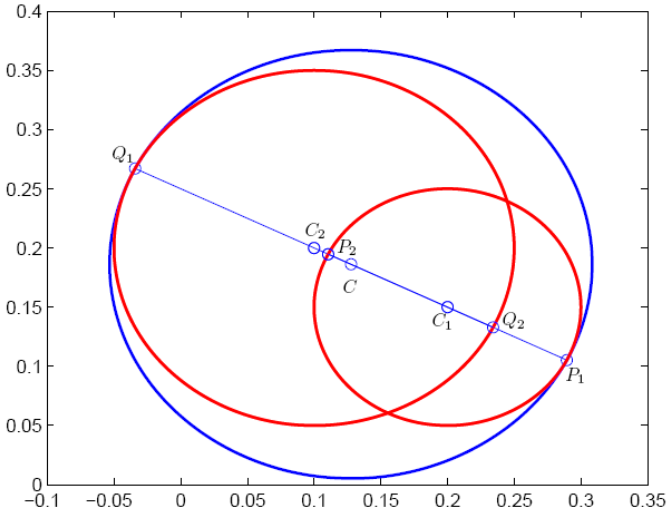

Fig. 1. The join hypersphere granule of two hypersphere granules

Fig. 1 shows the join process of the hypersphere granule $\boldsymbol{G}_1$ = [0.2 0.15 0.1] and the hypersphere granule $\boldsymbol{G}_2$ = [0.1 0.2 0.15]. The crosspoints of hypersphere granule $\boldsymbol{G}_1$ and the line crossing vector $\boldsymbol{C}_{12}$=[-0.1,0.05] are $\boldsymbol{P}_1$=[0.2894, 0.1053] and $\boldsymbol{P}_2$=[0.1106,0.1949]. The crosspoints of hypersphere granule $\boldsymbol{G}_2$ and the line crossing vector $\boldsymbol{C}_{21}$=[0.1 -0.05] are $\boldsymbol{Q}_1$=[-0.0342,0.2671] and $\boldsymbol{Q}_2$=[0.2342,0.1329]. According to algorithm1, the central vector and granularity of the join hypersphere granule $\boldsymbol{G}$ are $\boldsymbol{C}$=[0.1276,0.1862] and $\boldsymbol{R}$=0.1809, namely $\boldsymbol{G}$=[0.1276 0.1862 0.1809].

2.3 The Bottle-up Framework of GrC

For data set *S*, the granular computing clustering algorithms are proposed by the following steps. Firstly, the samples are used to form the atomic granule. Secondly, the threshold of granularity is introduced to conditionally union the atomic granules by the aforementioned join operation, and the granule set is composed of all the join granules. Thirdly, if all atomic granules are included in the granules of *GS*, the join process is terminated, otherwise, the second process is continued. The GrC algorithms are described as follows.

Suppose the atomic hypersphere granules induced by *S* are g_1, g_2, g_3, g_4, g_5, which have the same class labels. The training process can be described as the following tree structure shown in Fig. 2, leafs denote the atomic hypersphere granules, root denotes *GS* including its child nodes G_1, G_2, and g_3. G_1 is induced by join operation of child nodes g_1 and g_2, G_2 is the join hypersphere granule of g_4 and g_5, g_3 is the atomic hypersphere granule. The whole process of obtaining *GS* is the bottle-up process.

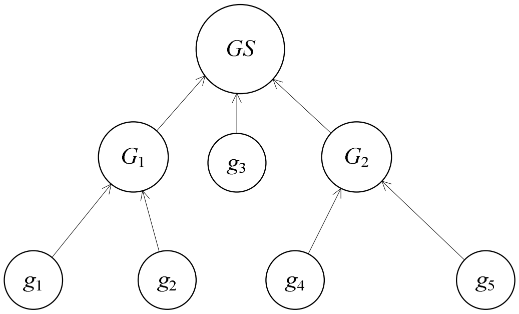

Fig. 2. The framework of GrCC

The GrC framework for the classification problems is described as algorithm 2.

```
Algorithm 2. GrC for the classification problems with n classes
Input: Training set S, threshold ρ of granularity
Output: Granule set GS and the corresponding class label lab
S1. initialize the granule set $GS=\varnothing$, $lab=\varnothing$
S2. $k$=1
S3. forming the subset $s$ of training set $S$ by the data with class label $k$
S4. $GS_t=\varnothing$
S5. $i$=1
S6. for the $i$th sample $\boldsymbol{x}_i$ in $s$, form the corresponding atomic granule $\boldsymbol{G}_i$
S7. $j$=1
S8. form the join granule $\boldsymbol{G}_i\vee\boldsymbol{G}_j$ of $\boldsymbol{G}_i$ and $\boldsymbol{G}_j\in GS_t$, if the granularity of $\boldsymbol{G}_i\vee\boldsymbol{G}_j$ is
less than or equal to ρ, then $\boldsymbol{G}_j=\boldsymbol{G}_i\vee\boldsymbol{G}_j$, else
S9. $j=j+1$
S10. if all the granularities of $\boldsymbol{G}_i\vee\boldsymbol{G}_j$ are greater than ρ, then $GS_t=GS_t\cup\{G_i\}$
S11. remove $\boldsymbol{x}_i$ from $s$ until $s$ is empty.
S12. $GS=GS\cup GS_t$
S13. if $k=n$, return $GS$, else $k=k+1$
```

3. DECISION TREE CLASSIFIERS BASED ON GRANULAR COMPUTING

A decision tree classifier creates a decision tree *t* for predicting the response *y* as a function of the predictors in the columns of *X*. *X* is an n-by-m matrix of predictor values. If *y* is a vector of *n* response values, decision tree performs regression. If *y* is a categorical variable, character array, or cell array of strings, decision tree performs classification. Either way, *t* is a binary tree where each branching node is split based on the values of a column of *X*. *NaN* values in *X* or *y* are taken as missing values, and observations with any missing values are not used in the fit.

We construct the decision tree classifiers based on granular computing from the following two aspects. Firstly, the aforementioned granular computing framework is performed by algorithm2 and the granule set is formed by the bottle up scheme. Secondly, decision tree classifiers are constructed by the top-down strategy. The decision tree classifier is described as follows.

```
Algorithm 3. Decision tree classifier
Input: Training set $S$ including $N$ attributes
Output: T
------------------------------------------------------------
S1. initialize the decision tree, $T=\varnothing$
S1. forming the root node, $T=T\cup\{Node\}$
S2. assigning the class label for Node
S3. if all the samples in S have the same class labels, or there is only one sample in S, then Node is a split leaf node
S4. partitioning the S for each attribute, and computing the Gini value of partition
S5. partitioning S into subset S1 and subset S2 according to the attribute with the minimal Gini value
S6. if the samples in subset S1 have the same class labels, the corresponding Node is joined into $T$, $T=T\cup\{Node\}$, and the procedure is terminated, otherwise steps from S1 to S5 are repeated
S7. if the samples in subset S2 have the same class labels, the corresponding Node is joined into $T$, $T=T\cup\{Node\}$, and the procedure is terminated, otherwise steps from S1 to S5 are repeated
```

For the training set S, the general framework of decision tree algorithms is described as follows.

```
Algorithm 4. Framework of DTCGrC
Input: Data set $S$ including $N$ attributes and $n$ class labels, threshold $\rho$ of granularity
Output: T
------------------------------------------------------------
S1. Performing the algorithm2, and obtaining the granule set and its class label set.
S2. forming decision tree by algorithm3
```

4. EXPERIMENTS

We verified the feasibility of DTCGrC by data set selected from UCI benchmark data set. The data set in 2-dimensional space is used to show the data and the achieved granules, and the data sets in *N*-dimensional space are used to verify the extension of DTCGrC. We evaluated the effectiveness of DTCGrC and decision tree classifier (DTC) using Intel(R) Core(TM) i5 CPU with 3.2GHz and 8 GB memory, running Microsoft Win7, and Matlab 2008.

In order to compare the DTCGrC with the decision tree classifier (DTC). The 10-fold cross validation is used to perform the two algorithms, namely, the data set is divided into 10 subsets of the same size, where 9 subsets are composed of training set, the rest subset is testing set, and DTCGrC and DTC are performed 10 times for a data set.

We mainly analyze and discuss DTCGrC compared with DTC from the testing accuracies and training time, the testing accuracies include minimal accuracy, maximal accuracy, mean accuracy, and standard deviation (std) of testing accuracies, the training time include minimal training time, maximal training time, mean training time, and std of training time.

Data sets (see Table 1) selected from Website (http://sci2s.ugr.es/keel/datasets.php) are used to evaluate the performances of DTCGrC. For the selection of parameter ρ, because the granularity is related to the size of attribute set, the parameters ρ for the selected data sets are set as follows. The performances are set from the maximal ρ to the minimal ρ with the step and listed in Table 2 compared with DTC.

Table 1. Classification problems

Data sets	Sizes	Attributes	Classes
Banknote	1372	4	2
Iris	150	4	3
Mammographic	830	5	2
Newthyroid	215	5	3
Segment	2300	19	7
Letter	20000	16	26
Skin	245057	3	2
Wilt	4839	5	2

Table 2. Parameters of DTCGrC for the selected data sets

Data sets	Maximal ρ	Minimal ρ	Step
Banknote	5	0	-0.1
Iris	5	0	-0.1
Mammographic	5	0	-0.1
Newthyroid	5	0	-0.1
Segment	100	0	-2
Letter	5	0	-0.1
Skin	5	0	-0.1
Wilt	5	0	-0.1

Table 3. Comparison of DTCGrC and DTC for the selected classification problems

Datasets	Algorithms	Testing accuracy (%)				Training time (s)			
		Min	Max	Mean	Std	Min	Max	Mean	Std
Banknote	DTCGrC	**96.3504**	**100.000**	**98.9072**	1.1534	0.0468	0.1248	0.0936	0.0294
	DTC	96.3504	99.2701	98.2513	**0.9838**	**0.0000**	**0.0312**	**0.0140**	**0.0089**
Iris	DTCGrC	93.3333	100.000	**97.3333**	**3.4427**	0.0000	0.0312	0.0125	0.0161
	DTC	93.3333	100.000	96.6667	3.5136	**0.0000**	**0.0156**	**0.0016**	**0.0049**

Datasets	**Algorithms**	**Testing accuracy (%)**				**Training time (s)**			
		Min	**Max**	**Mean**	**Std**	**Min**	**Max**	**Mean**	**Std**
Mammographic	DTCGrC	**76.0870**	**90.2439**	**84.1941**	4.2374	0.0312	0.0780	0.0530	0.0168
	DTC	72.8261	82.9268	79.4777	**2.8342**	**0.0000**	**0.0312**	**0.0172**	**0.0137**
Newthyroid	DTCGrC	**88.4615**	100.000	**96.4652**	**4.3066**	0.0000	0.0468	0.0234	0.0169
	DTC	76.9231	100.000	92.4542	6.9869	**0.0000**	**0.0312**	**0.0125**	**0.0143**
Segment	DTCGrC	**94.3723**	**98.7013**	**96.4069**	**1.3544**	0.2964	0.7800	0.6209	0.1282
	DTC	93.9394	97.4026	95.4113	1.4602	**0.1092**	**0.1248**	**0.1123**	**0.0066**
Letter	DTCGrC	**85.5058**	87.5692	**86.4072**	**0.6199**	11.4349	20.8417	15.1992	2.9905
	DTC	84.8012	**88.3241**	86.3679	1.0311	**2.2620**	**2.4336**	**2.3307**	**0.0551**
Skin	DTCGrC	**98.0126**	**99.9062**	**99.3503**	**0.5562**	14.0557	137.328	59.5284	42.3599
	DTC	92.0135	99.8450	98.8786	2.4272	**0.7956**	**1.0140**	**0.9251**	**0.0678**
Wilt	DTCGrC	**93.4959**	99.3789	**97.9417**	**1.7164**	0.7176	0.8268	0.7753	0.0435
	DTC	92.6829	99.3789	97.6534	1.9230	**0.0468**	**0.0780**	**0.0515**	**0.0105**

The besting testing accuracy is the selection index for classification problems. The performances are shown in Table 3 above. From the table, we can see the DTCGrC improved the testing accuracies from the aspects, such as the minimal, maximal, and mean testing accuracies. For the same classification problem, the std is used to measure the stability of algorithms, the smaller std represents the more stable algorithm. DTCGrC and DTC have the different stabilities for the different classification problems, DTCGrC is more stable than DTC, such as the classification problems iris, newthyroid, segment, letter, skin, and wilt, DTC is more stable DTCGrC, such as banknot and mammographic. The training time is used to measure the complexity of algorithms DTCGrC and DTC, DTC has the low complexity compared with DTCGrC because GrC is performed before construction of DTC in DTCGrC.

5. CONCLUSION

In this work, the novel decision tree classifiers are proposed based on granular computing. The novel classifiers include two stages, the first stage is to obtain the granule set by granular computing, the second stage is to form the decision tree classifiers for the obtained granule set. The experimental results show that the proposed classifiers improved the generalization ability compared with the traditional decision tree classifiers. The drawback of DTCGrC is the more training time compared with traditional decision tree classifiers. For classification algorithm, the application is very important, the paper discusses the improvement and verification of the performance of the improved decision tree classification algorithm. In the future work, we will applied the improved algorithm to computer vision field, such as image segmentation, image reconstruction.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Katz G, Shabtai A, Rokach L, Ofek N. ConfDTree: A statistical method for improving decision trees. Journal of Computer Science and Technology. 2014; 29(3):392-407.
2. Padua L, Schulze H, Matkovic K, Delrieux C. Interactive exploration of parameter space in data mining: Comprehending the predictive quality of large decision tree collections. Computers and Graphics. 2014; 41:99-113.
3. Lowd D, Davis J. Improving Markov network structure learning using decision trees. Journal of Machine Learning Research. 2014;15(1):501-532.
4. Guo JM, Liu YF, Chang JY, Lee JD. Fingerprint classification based on decision tree from singular points and orientation field. Expert Systems with Applications. 2014;41(2):752-764.
5. Chen MY. Comparing traditional statistics, decision tree classification and support vector machine techniques for financial bankruptcy prediction. Intelligent Automation & Soft Computing. 2012;18(1): 65-73.
6. Pedrycz W, Song M. A granulation of linguistic information in AHP decision-making problems. Information Fusion. 2014;17:93-101.
7. Pedrycz W. Associations among information granules and their optimization in granulation-degranulation mechanism of granular computing. International Journal of Fuzzy Logic and Intelligent Systems. 2013;13(4):245-253.
8. Papadakis SE, Kaburlasos VG, Papakostas GA. Two Fuzzy Lattice Reasoning (FLR) classifiers and their application for human facial expression recognition. Multiple-valued Logic and Soft Computing. 2014;22(4-6):561-579.
9. Kaburlasos VG, Papadakis SE, Papakostas GA. Lattice computing extension of the FAM neural classifier for human facial expression recognition. IEEE Transactions on Neural Networks and Learning Systems. 2013;24(10):1526-1538.
10. Jamshidi Y, Kaburlasos VG. gsaINknn: a GSA optimized, lattice computing knn classifier. Engineering Applications of Artificial Intelligence. 2014;35:277-285.
11. Kaburlasos VG, Pachidis T. A Lattice-computing ensemble for reasoning based on formal fusion of disparate data types, and an industrial dispensing application. Information Fusion. 2014;16:68-83.
12. Papakostas GA, Hatzimichailidis AG, Kaburlasos VG. Distance and similarity measures between intuitionistic fuzzy sets: A comparative analysis from a pattern recognition point of view. Pattern Recognition Letters. 2013;34(14): 1609-1622.

13. Wu S, Nagahashi H. Analysis of generalization ability for different AdaBoost variants based on classification and regression trees. Journal of Electrical and Computer Engineering; 2015. Available:http://www.hindawi.com/journals/jece/2015/835357/

Peer-review history:
The peer review history for this paper can be accessed here:
http://sciencedomain.org/review-history/11508