ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/324654835

Review of the Data Mining algorithms with proposed hybrid theorem for increasing accuracy

Conference Paper · April 2018

CITATIONS
0

READS
226

1 author:

Abeer D. Salman
Al-Maarif University
12 PUBLICATIONS **3** CITATIONS

SEE PROFILE

All content following this page was uploaded by Abeer D. Salman on 20 April 2018.

The user has requested enhancement of the downloaded file.

Review of the Data Mining algorithms with proposed hybrid theorem for increasing accuracy

Asst. Lecturer. Abeer Dawood Salman (MSc)
Computer Engineering Techniques
Al-Maarif University College
abeer.dawood@auc-edu.org

Abstract

Databases are rich with unobserved information that can be used for intelligent decision making. Data mining is defined as an automatic or semi-automatic exploration and analysis of large quantities of data so as to discover meaningful patterns. One of the important goals of data mining is predictive objectives (e.g. classification, regression, anomalies/outliers detection), carried out by using a part of the variables to predict one or more of the other variables. Classification has many applications including fraud detection, target marketing, manufacturing, performance prediction, and medical diagnosis.

In this paper, the main ideas of classification are introduced and the basic techniques of data classification, such as decision tree, and Bayesian classifiers, will be we learned and the paper survey the criteria that are used to evaluate and compare different classifiers. Several measures of accuracy are given as well as techniques for obtaining reliable accuracy. Hybrid theorem for increasing classifier accuracy are presented at the end of this paper.

مراجعة لخوارزميات تعدين البيانات مع الخوارزمية الهجينة المقترحة لزيادة الدقه

المدرس المساعد. عبير داود سلمان (ماجستير)
هندسة تقنيات الحاسوب
كلية المعارف الجامعة

الخلاصة

ان قواعد البيانات تحتوي على العديد من المعلومات المضمنة التي تستخدم في علمية اتخاذ القرار بشكل ذكي. يعرف تنقيب البيانات بانه عملية الاستكشاف والتحليل التلقائي او شبه التلقائي لكميات كبيرة من البيانات من اجل اكتشاف الانماط ذات المعنى. واحد من اهم اهداف تنقيب البيانات هو تخمين الاهداف (مثل: التصنيف و الانحدار، و كشف الشذوذ اوالقيم المتطرفة بالبيانات)، المتحقق عن طريق استخدام جزء من المتغيرات لتخمين واحد او اكثر من المتغيرات الاخرى. للتصنيف تطبيقات واسعة منها (الكشف عن الاحتيال، التسويق، التنبأ بالاداء، التصنيع، والتشخيص الطبي) .

في هذه الورقة البحثية سوف نبدأ بتقديم الافكار الاساسية في التصنيف و سوف نتعلم التقنيات الاساسية المستخدمة لتصنيف البيانات مثل (شجرة القرار وتصنيف بايزن). وسوف نقوم بمسح المعايير المستخدمة لتقييم ومقارنة مختلف المصنفات.عدة مقاييس للقياس الدقة سوف تقدم في هذا البحث بالاضافة الى التقنيات المستخدمة للحصول على دقة موثوقة. الطريقة المهجنة التي استخدمت لزيادة دقة المصنف سوف تعرض في نهاية الورقة البحثية.

1. Introduction

Database contains massive amount of data that are collected and stored from all over place. There is precious information and knowledge "hidden" in such databases. Mining data inside the DB in order to extract the information cannot be done without utilizing automatic methods, for this many algorithms are made for this purpose. Different methodologies are founded to deal with this trouble like: ***classification***, ***association rule, clustering***, etc **[1]**.

This research deal with classification where there are numerous techniques used in classification like: Decision Tree based Methods, Bayesian Theorem, Support Vector Machines, Memory based reasoning, and Neural Networks. This paper concentrated on the two classification methods, and shown the main difference between them, their benefits and drawbacks also explained. Finally hybrid technique are presented.

2. Data Mining Classification

Classification defined as a methodology that predicting some output based on a given input. The classification process divided into two parts (training and prediction).

In the **training phase**, there are a set of attributes and the respective output (usually called prediction attribute), any classification algorithm tries to find relations between these attributes that assistance to predict the goal. For instance, the training set of the medicinal database has patient information recorded beforehand with relevant outcomes (see Table 1).

While in the **prediction stage**, new data are given to the algorithm contains the similar set of attributes above, except prediction attribute. The function of the algorithm produces the goal by analyzing the input. Whenever the prediction was accurate, the algorithm is good. For the training set in (Table 1) the objective is to decide if the patient had a heart problem or not (see Table 2).

Table 1: Training Set			
Age	Heart Rate	Blood Pressure	Heart Problem
66	79	151/71	Yes
35	82	111/77	No
70	77	109/66	No

Table 2: Prediction Set			
Age	Heart Rate	Blood Pressure	Heart Problem
44	96	147/38	?
64	59	107/63	?
81	77	150/66	?

To express the knowledge, classification algorithms normally use ***prediction rules*** that is expressed in the form of **IF-THEN** rules. For example:

IF (Age=64 AND Heart rate>70) OR (Age>60 AND Blood pressure>140/70) THEN Heart problem=yes **[1]**.

3. Decision Tree based Methods

In General a decision tree is a flowchart-like tree structure, the tree has three types of nodes (see Figure 1):

- ***A Root node***: that has no arriving edges and zero or more leaving edges.
- ***An Internal node***: each of which has exactly one arriving edges and two or more leaving edges.
- ***A Leaf (terminal) nodes***: each of which has exactly one arriving edge and no leaving edges **[2]**.

In a decision tree class label is represented by each leaf node. The other nodes (root and internal nodes) represent attributes test conditions to separate records that have various characteristics. The examples of the algorithms that are based on decision tree in their classification are ID3, C4.5, C5.0, CART, etc. These algorithms are distinguished by high speed in classification, ability to learn strongly and simple construction features. Decision tree algorithms are

considered greedy approach in which decision trees are built in a top-down recursive, there is no-backtracking, It begins with a training set of attributes and their associated class labels, and the training set is continuously partitioned into smaller subsets as the tree is being constructed **[3]**.

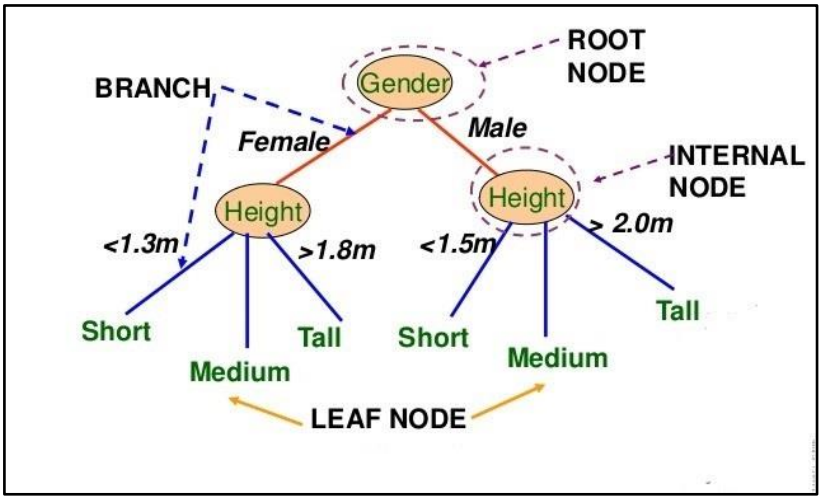


Figure 1 : Decision Tree Diagram

Splitting criteria

To form Tree, the node must be split depending on some splitting criteria that are used by all decision tree algorithms. The motivations behind these measures is to determine the best way to split the records **[3] [4]**. Some of these criteria are:

- ***Entropy***: The amount of information in an attribute is computed by entropy

$$Entropy(a) = -\sum P(i/a) log_2 P(i/a) \quad \textbf{(1)}$$

- ***Gini Index***: measures the variation between the probability distributions of the target attribute's values.

$$Gini\ Index = 1 - \sum [P(i/a)]^2 \quad \textbf{(2)}$$

- ***Information Gain***: The difference between the entropy of the parent node and child node is measured by this criterion.

$$Info\ Gain(\nabla) = Entropy(parent\ node) - Entropy(child\ node) \quad \textbf{(3)}$$

- ***Gain Ratio***: It used in order to normalize the information gain as follows:

$$Gain\ Ratio = Information\ gain(\nabla)/Entropy \quad \textbf{(4)}$$

- ***Twoing***: This criterion is calculated as follows:

$$Twoing(a) = \frac{P_L P_R}{4}\left(\sum\left(\left|P\left(\frac{i}{a_L}\right) - P\left(\frac{i}{a_R}\right)\right|\right)\right)^2 \quad \textbf{(5)}$$

Where, p (i/a) denotes the section of records belonging to class i at a given node a.

Stopping Criteria

The splitting procedure will stop when one of the following stopping Criteria below reached by the classification algorithm **[4]**:

1. All values in the training set belong to the same class.
2. The maximum depth of the tree has been reached.
3. The number of status of the child node is less than the minimum number of status for parent nodes.
4. There is a threshold greater than the best splitting criteria.

4.1 Advantages and Disadvantages of using Decision Trees

Decision trees in general have several important benefits **[5] [6]*:***

1) The same trees will produce when doing transformations of variables.

2) Decision tree has high Efficiency, where you can rapidly express complex problems obviously, and when new information becomes available you can easily modify a decision tree.

3) Nonprofessional users can easily follow decision trees, even it was compacted for reason it's called "self-explanatory". For increasing understandability decision trees are changed into set of rules.

4) Decision trees can handle various kind of attributes like nominal and numerical.

5) Even you have datasets that contain mistakes or having missing value decision trees can manage it.

The disadvantages of Decision trees are* [5] [6]*:

1) The algorithms of decision tree for example ID3 and C4.5 require that the target's values only discrete.
2) As decision trees depend on the “divide and conquer” process, therefor the performance of it is well if the attributes are independent, but if there are many relevant features, the performance will decrease.
3) It has long training time.
4) The decision trees are sensitive to the training set, to irrelevant attributes and to noise because it has the greedy feature.

The algorithms of Decision tree

Here comparison between basic characteristic of three types of decision tree algorithms as follow **[4]**:

Table3: basic characteristic of decision tree algorithms

Name	Splitting Criteria	Attribute type	Missing values	Pruning Strategy
ID3	Use Information Gain	Deals with only Categorical value	Do not handle missing values.	No pruning
CART	Use Towing Criteria	Deals with both Categorical and Numeric value	Address missing values.	Cost-Complexity pruning
C4.5	Utilize Gain Ratio	Deals with both Categorical and Numeric value	Address missing values.	Error Based pruning

4.2 A basic decision tree algorithm is summarized below.

The input parameters are the training records E and the attribute set F **[7]**.

Algorithm1: decision tree induction algorithm
TreeGrowth (E,F)
1. **if** stopping_cond(E,F)=*true* **then**
2. *leaf* = creatNode().
3. *leaf.labe* = classify(E).
4. return *leaf*.
5. **else**
6. *root* = creatNode().
7. *root.test_cond*=find_best_split(E,F).
8. let V = {*v*|*v* is possible outcome of root.test_cond}.
9. **for** each $v \in V$ **do**
10. E_V={ *e* | *root.test_cond*(*e*) = *v* and $e \in E$}.
11. *child*=TreeGrowth(E_V,F)
12. add *child* as descendent of *root* and label the edge (root →child) as *v*.
13. **end for**
14. **end if**
15. **return** root

Example of Decision Tree:

Draw the Decision Tree using ID3 for the following table: The instance is **Class [8]**

	Gender	Type	Class
1	F	1	A
2	F	2	B
3	M	2	A
4	M	2	A
5	F	3	B
6	M	2	A
7	M	3	A

$$\boldsymbol{I(class)} = \sum -\boldsymbol{P}\log_2 \boldsymbol{P_i}$$

$$\boldsymbol{Entropy(a)} = \sum \frac{\boldsymbol{c_i}}{\boldsymbol{c}} \times \boldsymbol{I(c)}$$

No of Class A=5

No of Class B=2

- $\boldsymbol{I(Class)} = \left[\left(-\frac{5}{7}\log_2\frac{5}{7}\right) + \left(-\frac{2}{7}\log_2\frac{2}{7}\right)\right] = 0.863117$
- $\boldsymbol{Entropy(Gender)} = \frac{3}{7}\left[\left(-\frac{1}{3}\log_2\frac{1}{3}\right) + \left(-\frac{2}{3}\log_2\frac{2}{3}\right)\right] + \frac{4}{7}\left[\left(-\frac{4}{4}\log_2\frac{4}{4}\right)\right] = 0.39355$
- $\boldsymbol{Entropy(Type)} = \frac{1}{7}\left[\left(-\frac{1}{1}\log_2\frac{1}{1}\right)\right] + \frac{4}{7}\left[\left(-\frac{3}{4}\log_2\frac{3}{4}\right) + \left(-\frac{1}{4}\log_2\frac{1}{4}\right)\right] + \frac{2}{7}\left[\left(\left(-\frac{1}{2}\log_2\frac{1}{2}\right) + \left(-\frac{1}{2}\log_2\frac{1}{2}\right)\right)\right] = 0.74925$
  - $\boldsymbol{Gain(Gender)} = I(Class) - EGender)$

$= 0.863117 - 0.39355 = 0.469567$

- $\boldsymbol{Gain(Type)} = I(Class) - E(Type)$

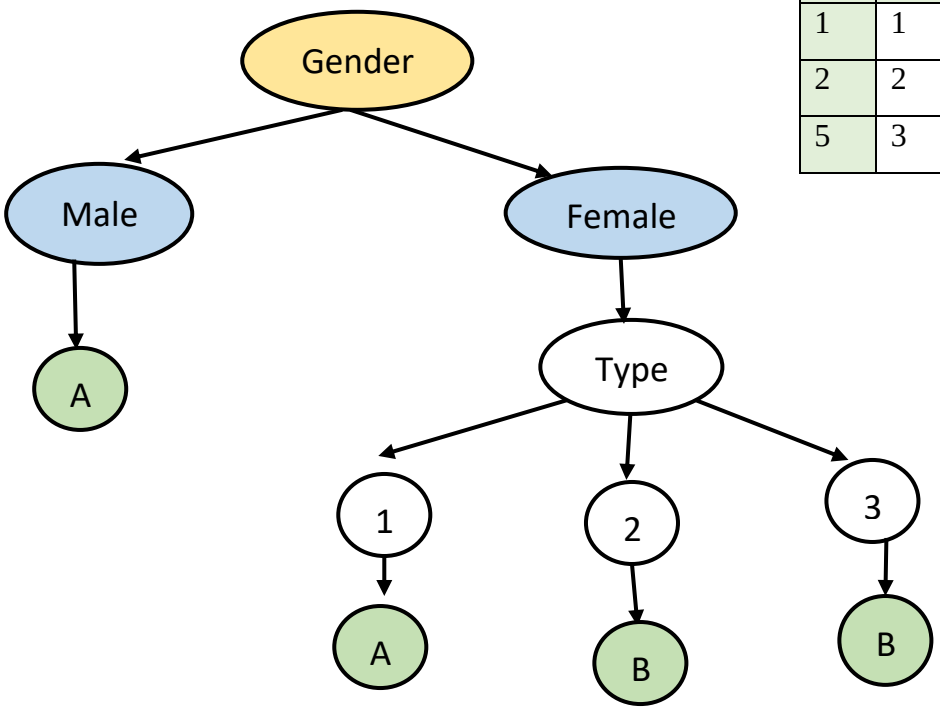
  $= 0.863117 - 0.74925 = 0.113867$

The largest gain is Gender therefor it become the root of the tree. We will show that when the Gender was **Male** the class was **A** therefor this is decision, and when Gender was **Female** class was between **A** and **B**, therefore we will split the DB gain without **Gender** attribute to show the final result.

$$I(Class) = \left[\left(-\frac{2}{3}\log_2\frac{2}{3}\right) + \left(-\frac{1}{3}\log_2\frac{1}{3}\right)\right] = 0.918295$$

$$Entropy(Type) = \frac{1}{3}\left[\left(-\frac{1}{1}\log_2\frac{1}{1}\right)\right] + \frac{1}{3}\left[\left(-\frac{1}{1}\log_2\frac{1}{1}\right)\right] + \frac{1}{3}\left[\left(-\frac{1}{1}\log_2\frac{1}{1}\right)\right] = 0$$

No.	Type	Class
1	1	A
2	2	B
5	3	B



Prediction Rules:

1) **IF** Gender is Male **THEN** the class is A
2) **IF** Gender is Female **AND** Type is 1 **THEN** the class is A
3) **IF** Gender is Female **AND** Type is 2 **THEN** the class is B
4) **IF** Gender is Female **AND** Type is 3 **THEN** the class is B

4. Bayes Classification Methods

The definition of Bayesian is statistical classifiers that can predict the probabilities of class membership, such as the probability that a given tuple belongs to a specific class. When Bayesian classifiers apply to large databases, it has given accuracy results with high speed.

Simple Bayesian classifier known as the *naïve Bayesian classifier*. *Naïve* assume, that the effect of the value of attribute on a given class is independent of the values of the other attributes. This hypothesis is called *class conditional independence* **[3]**.

Bayes Classifiers

Bayesian classifiers based on **Bayes theorem**, which says

$$P(c_j|d) = \frac{P(d|c_j).P(c_j)}{P(d)} \quad \textbf{(6)}$$

- $p(c_j \mid d)$ = probability of case d being in class c_j.
- $p(d \mid c_j)$ = probability of appearance of d in given class c_j.
- $p(c_j)$= probability of occurrence of class c_j in database
- $p(d)$= probability of instance d occurring. This value is ignored because, is the same for all classes **[3]**.

To simplify the job, **naïve Bayesian classifiers** take the distributions of attributes independently:

$$P(d|c_j) = \prod_{n=1}^{n} P(d_K|c_j) \quad \textbf{(7)}$$

$$P(d|c_j) = P(d_1|C_j) \times (d_2|C_j) \times (d_n|C_j)$$

The probability of class *cj* generating instance *d*, equals….

The probability of class *cj* generating the observed value for feature 1, multiplied by..

The probability of class *cj* generating the observed value for feature 2, multiplied by..

5.1 Advantages/Disadvantages of Naïve Bayes

The advantages of Bayesian stratiges are* [6][9]*:

1) The training and classifying process in Bayesian classifier is done quickly.
2) Handles many sorts of data like real, discrete, and streaming
3) Conceptually very easy to understand.
4) By removing the irrelevant features, it can enhance the classification performance.
5) Bayesian classifier does not require large amounts of data before beginning learning process.
6) It can make the decisions rapidly because the fact that it does not require high computations.
7) Compare to all other classifiers, Bayesian has the minimum error rate.

The weakness of Bayesian methods are* [6][9]*:

1) Assumes independence of features
2) Naive Bayes classifier requires a very large number of records to get great outcomes.
3) Compared to other classifiers on some dataset, it considered less accuracy.

Example of naïve Bayesian classifiers:

If we need to classify a person named "*drew*" as male or female classes. We need to know the probability of being "*drew*" male *p*(male| *drew*) and the probability of being female *p*(female| *drew*) the largest value decide the class**.** The algorithm is trained on the database contain some attributes with their values that help the classifier in the classification process **[10]**. The steps below show the naïve Bayesian classifier:

Name	Over 170CM	Eye	Hair length	Gender
Drew	No	Blue	Short	Male
Claudia	Yes	Brown	Long	Female
Drew	No	Blue	Long	Female
Drew	No	Blue	Long	Female
Alberto	Yes	Brown	Short	Male
Karin	No	Blue	Long	Female
Nina	Yes	Brown	Short	Female
Sergio	Yes	Blue	Long	Male

$$P(Gender|d) = \frac{P(drew|Gender).P(Gender)}{P(drew)}$$

$$P(\boldsymbol{male}|drew) = \frac{\frac{1}{3}.\frac{3}{8}}{\frac{3}{8}} = 0.125 \qquad P(female|drew) = \frac{\frac{2}{5}.\frac{5}{8}}{\frac{3}{8}} = 0.250$$

Prior probabilities of each class: $P(male) = 3/8, P(female) = 5/8$

The conditional probabilities of each attributes is computed independently as shown in the Figure below: for example:

P(Over 170_{cm}=yes | Gender=Male)=2/3. The rest in the same way

		Class: Gender	
		Male	Female
Over 170_{cm}	Yes	2/3=0.66	2/5=0.4
	No	1/3=0.33	3/5=0.6

		Class: Gender	
		Male	Female
Eye	Blue	2/3=0.66	3/5=0.6
	Brown	1/3=0.33	2/5=0.4

		Class: Gender	
		Male	Female
Hair length	Long	1/3=0.33	4/5=0.8
	Short	2/3=0.66	1/5=0.2

When the algorithm given such attributes

X=<over 170 cm=yes, eye=Blue, Hair length=long>

Equation (7) is applied to find the class whose greatest probability as follows:

$$\boldsymbol{P(drew|male)} = \left\{\frac{2}{3}.\frac{2}{3}.\frac{1}{3}\right\}.\frac{\mathbf{3}}{\mathbf{8}} = \mathbf{0.055} \qquad \boldsymbol{P(drew|female)} = \left\{\frac{2}{5}.\frac{3}{5}.\frac{4}{5}\right\}.\frac{\mathbf{5}}{\mathbf{8}} = \mathbf{0.12}$$

5. Comparison among Naive Bayes and Decision Tree techniques

Table 4 shows the comparison between Naive Bayes and Decision Tree Techniques **[6][2]**:

Table 4: Comparison between Naive Bayes and Decision Tree

	Parameter	Naïve Bayesian classifier	Decision Tree
1	**Deterministic/Non-Deterministic**	Non- Deterministic	Deterministic
2	**Effectiveness on**	Huge data	Large data
3	**Speed**	Fast	Faster
4	**Dataset**	It can deal with noisy data	It can deal with noisy data
5	**Accuracy**	To obtain good results, it requires a very large amount of records	High accuracy
6	**Application**	Text Classification, Spam Filtering	Pattern, Sequence, and Financial Recognition
7	**Data Types**	Numerical and categorical	Numerical and categorical
8	**Understandability**	Simple to understand and build	Simple to understand and generate

6. Hybrid Classification Techniques

As seen above both decision tree and Naïve Bayesian classifiers are efficient and commonly used for solving many classification problems in data mining. There is some problems arise with each type like the presence of noisy contradictory instances in the training set may cause creating decision tree suffers from overfitting conceptually the results will be less accurate.

For example: If there is a training dataset has two classes, and after applying naive Bayes classifier by calculating P(*Class*|*D*) for each instance based on prior and class conditional probabilities of training dataset, we have found some instances belong to "Class = X", but in the training dataset they are labeled as "Class = Y". To solve this problem hybrid DT algorithm and a naive Bayes

classifier are employed to eliminate the training set D from noisy (**misclassified**). After that a Decision Tree is built via utilizing the updated training dataset with apply the same steps mentioned in Algorithm1 **[11]**.

Algorithm 2: Hybrid decision tree and naive Bayes
Input: D = {x1,x2, . . . ,xn} // Training dataset, D, which contains training instances and their class labels.
Output: D without noise instance.

```
  for each class, Ci∈D, do
      Find the prior probabilities, P(Ci).
   end for
   for each attribute value, Aij∈D, do
      Find the class conditional probabilities, P(Aij|Ci).
   end for
   for each training instance, xi∈D, do
      Find the posterior probability, P(Ci|xi)
         if xi is misclassified, do
              Remove xi from D;
         end if
  end for
T = ∅;
Determine best splitting attribute;
T = Create the root node and label it with the splitting attribute;
T = Add arc to the root node for each split predicate and label;
for each arc do
      D = Dataset created by applying splitting predicate to D;
      if stopping point reached for this path,
        T' = Create a leaf node and label it with an appropriate class;
      else
        T' = DTBuild(D);
       end if
      T = Add T' to arc;
end for
```

7. Reference

[1] Fabricio voznika, Leonardo Viana, "Data Mining Classification", Springer, 2001.

[2] Bhavesh Patankar, Vijay Chavda, "A Comparative Study of Decision Tree, Naive Bayesian and k-NN Classifiers in Data Mining", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 12, December 2014.

[3] Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining Concepts and Techniques", Third Edition, Morgan Kaufmann Publishers, 2012.

[4] Sonia Singh, Priyanka Gupta," COMPARATIVE STUDY ID3, CART AND C4.5 DECISION TREE ALGORITHM: A SURVEY", International Journal of Advanced Information Science and Technology (IJAIST), Vol.27, No.27, July 2014

[5] Cristina Petri," Decision Trees", Cluj Napoca, 2010.

[6] Sayali D. Jadhav, H. P. Channe, "Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques", International Journal of Science and Research (IJSR), 2013.

[7] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, "Introduction to Data Mining", First Edition, 2005.

[8] George F. Luger, William A. Stubblefield, "Artificial Intelligence Structures and Strategies for Complex Problem Solving", Third Edition, 1998.

[9] Ahmad Ashari, Iman Paryudi, A Min Tjoa, "Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 4, No. 11, 2013.

[10] Eamonn Keogh UCR, "Naïve Bayes Classifier" http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

[11] Dewan Md. Farid, Li Zhang ,Chowdhury Mofizur Rahman, M.A. Hossain, Rebecca Strachan, "Hybrid decision tree and naive Bayes classifiers for multi-class classification tasks", Expert Systems with Applications, 2014

View publication stats