ISSN: 2321-7782 (Online)
Volume 3, Issue 3, March 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Improved Classification Approach by using FCM and CART in Data Mining

B.Sujatha
Asst. Professor
Annamacharya Institute of Technology and Science
India.

Abstract: ***Data Mining is a field of search and researches of data. Mining the data means fetching out a piece of data from a huge data block. The basic work in the data mining can be categorized in two subsequent ways. One is called classification and the other is called clustering. Although both refers to some kind of same region but still there are differences in both the terms. The classification of the data is only possible if you have modified and identified the clusters. In the presented research paper, our aim is to find out the maximum number of clusters in a specified region by applying the area searching algorithms. Classification is always based on two things. a)The area which you choose for the classification that is the cluster region .b)The kind of dataset which you are going to apply on the selected region .To increase the accuracy of the searching technique, any one would need to focus on two things . a)Whether the data set has been cauterized in proper manner or not .b)If the clusters are defined , whether they fit into the appropriate classified area or not .***

Keywords: Data Mining, FCM, CART, KDD, SVM - Algorithm.

I. INTRODUCTION

With the enormous amount of data stored in files, databases, and other repositories, it is increasingly important, if not necessary, to develop powerful means for analysis and perhaps interpretation of such data and for the extraction of interesting knowledge that could help in decision-making. Data Mining, also popularly known as Knowledge Discovery in Databases (KDD), refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases. Data mining refers to extracting or "mining" knowledge from large amounts of data.

Classification (technique to analyses the *frequent item sets*) is one of the major fields in the area of extracting knowledge from vast data. A *frequent item set* typically refers to a set of items that frequently appear together in a transactional data set, such as milk and bread (Han & Kamber, 2001). In this paper, we will briefly review about data mining, its architecture, functionalities, classification, methods of classification etc. We are in an age often referred to as the information age. In this information[1] age, because we believe that information leads to power and success, and thanks to sophisticated technologies such as computers, satellites, etc., we have been collecting tremendous amounts of information. Initially, with the advent of computers and means for mass digital storage, we started collecting and storing all sorts of data, counting on the power of computers to help sort through this amalgam of information. Unfortunately, these massive collections of data stored on disparate structures very rapidly became overwhelming. This initial chaos has led to the creation of structured databases and database management systems (DBMS). The efficient database management systems have been very important assets for management of a large corpus of data and especially for effective and efficient retrieval of particular information from a large collection whenever needed. The proliferation of database management systems has also contributed to recent massive gathering of all sorts of information. Today, we have far more information than we can handle: from business transactions and scientific data, to satellite pictures, text reports and military intelligence. Information retrieval is simply not enough anymore for decision-making. Confronted with huge collections of data, we have now created new needs to help us make better managerial

choices. These needs are automatic summarization of data, extraction of the "essence" of information stored, and the discovery of patterns in raw data.

II. FUZZY C – MEANS CLUSTERING ALGORITHM(FCM)

Most analytical fuzzy clustering algorithms (and also all the algorithms presented in this chapter) are based on optimization of the basic *c*-means objective function, or some modification if it. Hence we start our discussion with presenting the fuzzy *c*means functional.

The Fuzzy c-Means Functional

A large family of fuzzy clustering algorithms is based on minimization of the *fuzzyc-means* functional formulated as (Dunn, 1974; Bezdek, 1981): $J(\mathbf{Z};\mathbf{U},\mathbf{V}) = \_ci=1\_Nk=1(\mu ik)m\_\mathbf{z}k - \mathbf{v}i\_2\mathbf{A}$

where

$\mathbf{U} = [\mu ik] \in Mfc$ is a fuzzy partition matrix of $\mathbf{Z}$,

$\mathbf{V} = [\mathbf{v}1, \mathbf{v}2, \ldots, \mathbf{v}c]$, $\mathbf{v}i \in \mathrm{R}n$ is a vector of *cluster prototypes* (centers), which have to be determined,

$D2\,ik\mathbf{A} = \_\mathbf{z}k - \mathbf{v}i\_2$

$\mathbf{A} = (\mathbf{z}k - \mathbf{v}i)T\mathbf{A}(\mathbf{z}k - \mathbf{v}i)$

is a squared inner-product distance norm, and

$m \in [1, \infty)$ is a parameter which determines the fuzziness of the resulting clusters. The value of the cost function can be seen as a measure of the total variance of $\mathbf{z}\,k$ from $\mathbf{v}i$.

The Fuzzy c-Means Algorithm

The minimization of the *c*-means functional represents a nonlinear optimization problem that can be solved by using a variety of methods, including iterative minimization, simulated annealing or genetic algorithms. The most popular method is a simple Picard iteration through the first-order conditions for stationary points of known as the fuzzy *c*-means (FCM) algorithm.

The stationary points of the objective function can be found by adjoining the constraint to J by means of Lagrange multipliers:

$\bar{J}(\mathbf{Z};\mathbf{U},\mathbf{V},\lambda) = \_ci=1\_Nk=1(\mu ik)mD2ik\mathbf{A} + \_Nk=1\,\lambda k$

$/\_ci=1\,\mu ik - 10$,and by setting the gradients of $\bar{J}$ with respect to $\mathbf{U}$, $\mathbf{V}$ and λ to zero. It can be shown

that if $D2ik\mathbf{A} > 0$, $\forall i, k$ and$m > 1$, then $(\mathbf{U},\mathbf{V}) \in Mfc \times \mathrm{R}n \times c$ may minimize only if

$\mu ik = 1\_cj=1(Dik\mathbf{A}/Djk\mathbf{A})2/(m-1)$

, $1 \le i \le c$, $1 \le k \le N$,

and

$\mathbf{v}i = N\_k=1(\mu ik)m\mathbf{z}kN\_k=1(\mu ik)m$; $1 \le i \le c$.

This solution also satisfies the remaining constraints and first-order necessary conditions for stationary points of the functional. The FCM terates through and Sufficiency of and the convergence of the FCM algorithm is proven in (Bezdek, 1980). Note that gives $\mathbf{v}i$ as the weighted mean of the data items that belong to a cluster, where the weights are the membership degrees. That is why the algorithm is called "*c*-means".

Some remarks should be made:

1. The purpose of the "if . . . otherwise" branch at Step 3 is to take care of a singularity that occurs in FCM when$Dis\mathbf{A}$ = 0for some $\mathbf{z}k$ and one or more cluster prototypes $\mathbf{v}s$, $s \in S \subset \{1, 2, \ldots, c\}$. In this case, the membership degree in cannot be computed. When this happens, 0 is assigned to each μik, $i \in \bar{} S$ and the membership is distributed arbitrarily among μsj subject to the constraint $\_ s \in S \mu sj = 1, \forall k$.

2. The FCM algorithm converges to a *local* minimum of the *c*-means functional.Hence, different initializations may lead to different results.

3. While steps 1 and 2 are straightforward, step 3 is a bit more complicated, as a singularity in FCM occurs when $Dik\mathbf{A} = 0$ for some $\mathbf{z}k$ and one or more $\mathbf{v}i$.When this happens (rare in practice), zero membership is assigned to the clusters

Algorithm 4.1 Fuzzy *c*-means (FCM).

Given the data set **Z**, choose the number of clusters $1 < c < N$, the weighting exponent $m > 1$, the termination tolerance $(> 0$ and the norm inducing matrix **A**.Initialize the partition matrix randomly, such that $\mathbf{U}(0) \in Mfc$.

Repeat for $l = 1, 2, \ldots$

Step 1: Compute the cluster prototypes (means):

$\mathbf{v}(l)i = N\_k=1\_\mu(l-1)ik\_m\mathbf{z}kN\_k=1\_\mu(l-1)ik\_m, \ 1 \le i \le c$.

Step 2: Compute the distances:

$D2ik\mathbf{A} = (\mathbf{z}k - \mathbf{v}(l)i)T\mathbf{A}(\mathbf{z}k - \mathbf{v}(l)i), \ 1 \le i \le c, \ 1 \le k \le N$.

Step 3: Update the partition matrix:

for $1 \le k \le N$ if $Dik\mathbf{A} > 0$ for all $i = 1, 2, \ldots, c$ $\mu(l)ik = 1\_cj=1(Dik\mathbf{A}/Djk\mathbf{A})2/(m-1)$,otherwise $\mu(l)\ ik = 0$ if $Dik\mathbf{A} > 0$, and $\mu(l)ik \in [0, 1]$ with $\_c\ i=1\ \mu(l)ik = 1$.**until** $_\mathbf{U}(l) - \mathbf{U}(l-1)_ < (.$for which $Dik\mathbf{A} > 0$ and the memberships are distributed arbitrarily among the clusters for which $Dik\mathbf{A} = 0$, such that the constraint in is satisfied.

III. Implementation of 5.0 Algorithm on Provided Data

The data obtained from the excel sheet has been used as a source of data in paper so as to predict the out come of the student in the university exam. [3] However a slight modification has been done in the same data for a better prediction.

BUILDING CLASSIFICATION TREES

In the previous section we saw that the construction of a classification tree starts with performing good splits on the data. In this section we define what such a good split is and according to these splits.

TREE CONSTRUCTION

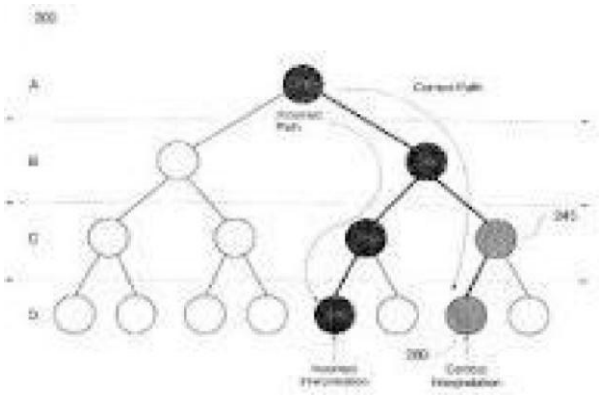
Building a classification tree starts at the top of the tree with all the data. For all the attributes the best split of the data must be computed. Then the best splits for each of the attributes are compared. The attribute with the best split wins. The split will be executed on the attribute with the best value of the best split (again we consider binary trees). The data is now separated to the corresponding branches and from here the computation on the rest of the nodes will continue in the same manner. Tree construction will finish when there is no more data to separate or no more attributes to separate them by Over fitting and Pruning.

If possible we continue splitting until all leaf nodes of the tree contain examples of a single class. But unless the problem is deterministic, this will not result in a good tree for prediction. We call this over fitting. The tree will be focused too much on the training data. To prevent over fitting we can use stopping rules; stop expanding nodes if the impurity reduction of the best split

is below some threshold. A major disadvantage of stopping rules is that sometimes, first a weak (not weaker) split is needed to be able to follow up with a good split. This can be seen in building a tree for the XOR problem pract DM. Another solution is pruning. First grow a maximum-size tree on the training sample and then prune this large tree. The objective is to select the pruned sub tree that has the lowest true error rate. The problem is, how to find this pruned sub tree .There are two pruning methods we will use in the tests, cost- complexity pruning [1] and [5] and reduced-error pruning [3]. In the next two paragraphs we will explain how the two pruning methods work and finish with a concrete example of the pruning process. Cost-complexity pruning The basic idea of cost-complexity pruning is not to consider all pruned sub trees, but only those that are the "best of their kind" in a sense to be defined below. Let R(T) (T stands for the complete tree) denote the fraction of cases in the training sample that is misclassified by the tree T (R(T) is the weighted summed error of the leafs of tree T). Total cost Cfi(T) of tree T is defined as: Cfi(T) = R(T) + fi| ˜ T| (7) The total cost of tree T then consists of two components: summed error of the leafs R(T), and a penalty for the complexity of the tree fi| ˜ T|. In this expression ˜ T stands for the set of leaf nodes of T, | ˜ T| the number of leaf nodes and fi is the parameter that determines the complexity penalty: when the number of leaf nodes increases by one (one additional split in a binary tree), then the total cost (if R remains equal) increases with fi [4]. The value of fi can make a complex tree with no errors have a higher total cost than a small tree making a number of errors. For every value of fi there is a smallest minimizing subtree. We state the complete tree by Tmax. For a fixed value of fi there is a unique smallest minimizing subtree T(fi) of Tmax.

IV. CART ALGORITHM

Classification and Regression Trees is a classification method which uses historical data to construct so-called decision trees. Decision trees are then used to classify new data. In order to use CART we need to know number of classes a priori. CART methodology was developed in 80s by Breiman, Freidman, Olshen, Stone in their paper"Classification and Regression Trees" (1984). For building decision trees, CART uses so- called learning sample - a set of historical data with pre- assigned classes for all observations. For example, learning sample for credit scoring system would be fundamental information about previous borrows (variables) matched with actual payoff results (classes). Decision trees are represented by a set of questions which splits the learning sample into smaller and smaller parts. CART asks only yes/no questions. A possible question could be:"Is age greater than 50?" or "Is sex male?".[5] CART algorithm will search for all possible variables and all possible values in order to find the best split – the question that splits the data into two parts with maximum homogeneity. The processes then repeated for each of the resulting data fragments. Here is an example of simple classification tree, used by San Diego Medical Center for classification of their patients to different levels of risk:



In practice there can be much more complicated decision trees which can include dozens of levels and hundreds of variables. As it can be seen from figure 1.1, CART can easily handle both numerical and categorical variables. Among other advantages of CART method is its robustness to outliers. Usually the splitting algorithm will isolate outliers in individual node or nodes. An important practical property of CART is that the structure of its classification or regression trees is

invariant with respect to monotone transformations of independent. variables. One can replace any variable with its logarithm or square root value, the structure of the tree will not change.

CART methodology consists of tree parts:

1. Construction of maximum tree

2. Choice of the right tree size

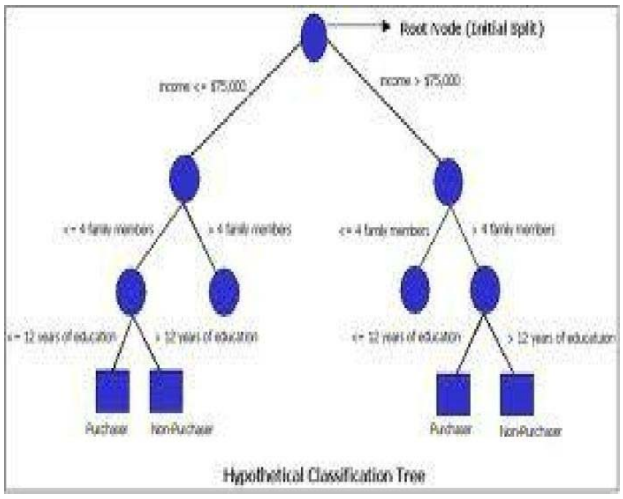
3. Classification of new data using constructed tree

Construction of Maximum Tree

This part is most time consuming. Building the maximum tree implies splitting the learning sample up to last observations, i.e. when terminal nodes contain observations only of one class. Splitting algorithms are different for classification and regression trees. Let us first consider the construction of classification trees.

V. CLASSIFICATION TREE

Classification trees are used when for each observation of learning sample we know the class in advance. Classes in learning sample may be provided by user or calculated in accordance with some exogenous rule. For example, for stocks trading project, the class can be computed as a subject to real change of asset price. Let tp be a parent node and tl,tr - respectively left and tight child nodes of parent node tp. Consider the learning sample with variable matrix X with M number of variables xj and N observations. Let class vector Y consist of N observations with total amount of K classes. Classification tree is built in accordance with splitting rule - the rule that performs the splitting of learning sample into smaller parts. We already know that each time data have to be divided into two parts with maximum homogeneity:



Hypothetical Classification Tree

where tp, tl, tr - parent, left and right nodes; xj - variable j; xRj best splitting value of variable xj . Maximum homogeneity of child nodes is defined by so-called impurity function i(t). Since the impurity of parent node tp is constant for any of the possible splits xj fi xRj , j = 1, . . . ,M, the maximum homogeneity of left and right child nodes will be equivalent to the maximization of change of impurity function fii(t): i(t) = i(tp) − E[i(tc)] where tc - left and right child nodes of the parent node tp. Assuming that the Pl, Pr - probabilities of right and left nodes, we get i(t) = i(tp) − Pli(tl) − Pri(tr) Therefore, at each node CART solves the following maximization problem: arg max xjfixRj, j=1,...,M [i(tp) − Pli(tl) − Pri(tr)] Equation implies that CART will search through all possible values of all variables in matrix X for the best split question xj < xRj which will maximize the change of impurity measure fii(t). The next important question is how to define the impurity function i(t).

VI. Proposed Work

In the current Industry of the data mining,the efficiency to increase the identification area has been a major task which has been getting implemented using different types of algorithm. According to the law of cycle, hundred percent efficiency is not possible but still the researchers have been trying to take maximum out of it. Different scientist has implemented their different types of parameters and criteria to figure out how effective the classification could be. Few names has left their mark in such kind of work with their popular proposed algorithms like CART, SVM C MEAN and many more. Our aim is to find a hybrid algorithm which can implement better result than the proposed algorithms till now. For this aim, we are going to combine two most effective results like CART AND CMEAN to implement a sophisticated architecture which has both the features like tree architecture and a mean square average algorithm. By combining two algorithms we would be definitely be able to use both the features and the results would be definitely better.

CMEAN to implement a sophisticated architecture which has both the features like tree architecture and a mean square average algorithm. By combining two algorithms we would be definitely be able to use both the features and the results would be definitely better.

VII. Conclusion

This paper concludes that the accuracy of classifying data can be improved by hybrid the feature of C mean and CART algorithm and their implementation on the MATLAB shows their resuls improvement in terms of accuracy compared to existing classifying algorithm. A direct algorithm of C-means method requires time proportional to the product of number of patterns and number of clusters identified. CART algorithm is a decision based algorithm which is used here with C mean algorithm in order to improve the efficiency of classifying data in terms of accuracy ,no of clusters formed to classify the data and the time taken to classify the data from an available vast amount of data. Features of C mean and CART are hybrid here and implemented over the Matlab software tool which generates graphical form of relationships between accuracy, no of clusters identified for classification and the time taken to classify data.

References

1. Leonid Churilov, Adyl Bagirov, Daniel Schwartz, Kate Smith and Michael Dally, -Proceedings of theWorkshop on Educational Data Mining at the 8th International Conference on Intelligent Tutoring Systems Jhongli, Taiwan.,2011.
2. Jiawei Han and Micheline Kamber 2 ed."Data mining concepts and Technique.
3. Bagirov et al.. Classification and Regression Trees. Chapman and Hall, New York, NY, 1984.
4. Huda Akil, Maryann E. Martone, David C Van Essen _. Multivariate versus univariate decision trees. Technical Report 92-8, Department of Computer Science, University of Massachusetts, Amherst, MA, 1992.
5. S. K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees Journal of Arti_cial Intelligence Research, 2:1{33, 1994.

