ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

TREE DATASET EXTRACTION USING HAC BASED ALGORITHM

G.B.Yogavarshini, V.Vasanthi, N.Deepa Asst.Prof.
B.E. Computer Science and Engineering
G.K.M. College of Engineering and Technology

DOI: 10.5281/zenodo.47029

ABSTRACT

The main objective of this project is to formulate a trouble-free ways of fetching data using TREE SET data structures and finding appropriate values and here we are using unproblematic apparent concepts of data mining which is an analytical process designed to explore enormous amounts of data typically business or market related. The project concludes by comparing the proposed quantifiers to a more traditional approach minimum spanning tree which is proficient to find the nodes within the best case. So there will be no need of long traverse and also Hierarchical Archimedean Copulas (HAC) helps to cluster and embed the nodes in different categories.

KEYWORDS: data structures, Data mining, generalized quantifiers, minimum spanning tree, Hierarchical Archimedean Copulas.

INTRODUCTION

Data Mining is the procedure of extracting hidden predictive information from large sets of data. From given databases, data mining technology generates new business opportunities by providing these capabilities:

Prediction of trends and behaviors:

Data mining automatically does the process of finding predictive information from databases. A best example of a predictive problem is targeted marketing which uses data on past to identify the targets leads to maximize return on investment. Other predictive problems include forecasting and identifying segments of a population.

Discovering previously unknown patterns:

Data mining tools goes through databases and identify previously hidden patterns in a single step. An example of hidden pattern discovering is analyzing of retail sales data to identify unrelated products that are generally purchased together. Other unknown pattern discovery problems include detecting crooked credit card transactions and identifying peculiar data that could represent data entry keying errors.

Databases may be larger in both depth and breadth:

More columns:

Analysts must usually reduce the number of variables they examine when doing direct analysis due to time constraints. High performance data mining allows users to examine the full depth of a database, without selecting a subset of variables.

More rows:

Larger samples earn lower estimation errors and variance which allow users to make assumptions about every small important segment of a population.

METHODOLOGY

TreeSet:

A set is a collection of objects in which the objects cannot be repeated.Being a Set,a TreeSet has the property that the elements present in set are arranged in ascending order. For each iteration of a TreeSet will always visit the elements of the set in ascending order.The way to determine the sorted order is, the objects in a set of type

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

TreeSet<T> should implement the interface Comparable<T> and ob1.compareTo(ob2) should be defined for any two objects ob1 and ob2 in the set. On the other hand,when the TreeSet is created an object of type Comparator<T> is provided as a parameter to the constructor. The Comparator make use of CompareTo() method to compare objects that are added to the set.The compareTo() considers two objects to be the same for comparing them even though the objects are not equal. So the TreeSet allows only one of those objects to remain in the set.The TreeSet implemented in such a way that the elements are stored similar to a binary sort tree. The data structure that is used is balanced that all the leaves of the tree are at the same distance from the root of the tree. The fact is that the TreeSet sorts its elements and removes duplicates.A TreeSet automatically eliminates the duplicates, and the iterator of the set automatically visits the items contained by the set in sorted order. An algorithm for TreeSet,

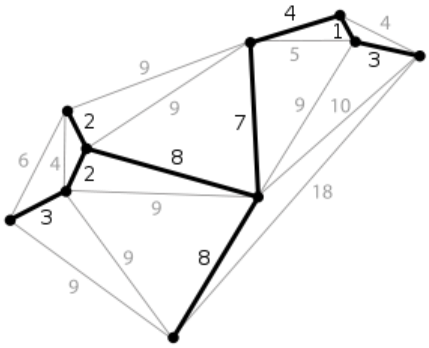
```
TreeSet set1 = new TreeSet();
set.addAll(coll1);
ArrayList list1 = new ArrayList();
list1.addAll(set1);
ArrayList list1 = new ArrayList( new TreeSet(coll1) );
 //coll1-collection of strings
```

This will make the sorted list of the elements of coll1 with no duplications.

Minimum Spanning Tree:

A minimum spanning tree is a spanning tree of a connected and undirected graph. It connects all the vertices together, which has the minimal total weighting for its edges. We can assign a weight to each edge, using that we can assign a weight to a spanning tree by computing the sum of the weights of the edges in that spanning tree. A single graph consists of many spanning trees. A Minimum Spanning Tree (MST) or Minimum Weight Spanning Tree is then a spanning tree with weight either less than or equal to the weight of spanning tree. In general undirected graph has a minimum spanning tree forest which is a union of minimum spanning trees for its connected components.

Figure1:*Example for MST*



A planar graph and its minimum spanning tree with each edge labeled with its weight which are roughly proportional to its length.

Optimal algorithm:

Seth Pettie and VijayaRamachandran have found a provably optimal deterministic comparison-based minimum spanning tree algorithm. The following are simplified description of the algorithm.

1. Assume r = \log \log \log n, where n is the number of vertices. All optimal decision trees on r vertices can be found in time O(n).
2. Optimal decision trees are used to find an MST for each component.
3. Graph are portioned into components with at most r vertices in each component with in time O(m).
4. It is possible to prove that the resulting graph has at most n/r vertices thus the graph is dense and use any algorithm which works on Dense graphs in time O(m).The runtime of all the steps in the algorithm is O(m), except for the step using the decision trees. We don't know the runtime of these step, but we know that it is optimal were no algorithm can do better than the optimal decision tree. Thus the algorithm has the peculiar property that it is optimal although its runtime complexity is unknown.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

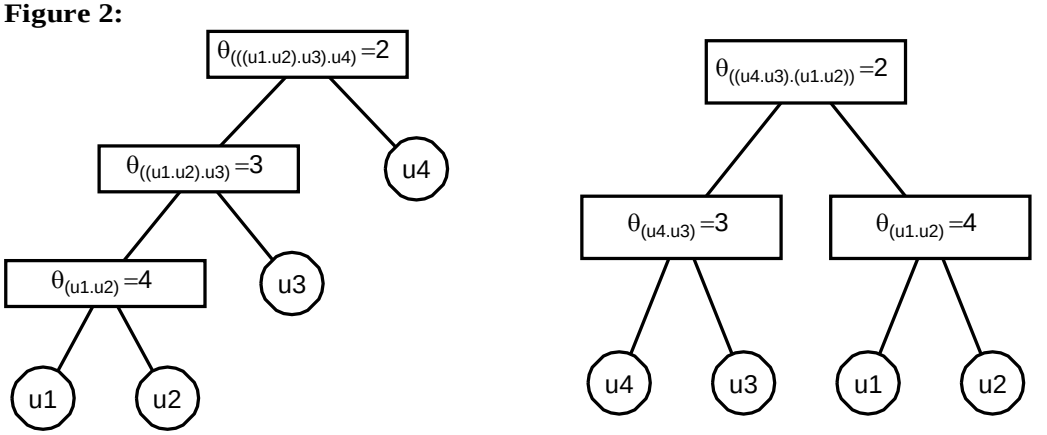
Hierarchical Archimedean Copula:

Let m $\in N$, $m \geq 2$ and (V, ε) be a tree with a root $v_1 \in V$ and m leaves, and all remaining nodes have at least 2 children; those nodes will be called forks. The large class of copula, which can describe tail dependency, non-elliptic, and most importantly, has close form representation,

$$C(u_1, \dots, u_d; \theta) = \emptyset_\theta\{\emptyset_\theta^{-1}(u_1) + \cdots + \emptyset_\theta^{-1}(u_d)\}, u_1, \dots, u_d \in [0,1], \quad (1)$$

Where $\varphi_\theta(.) \in L = \{\varphi_\theta : [0;\infty) \to [0,1] | \varphi_\theta(0) = 1], \varphi_\theta(\infty) = 0; (-1)^j \varphi^{(j)} \geq 0; j \in N\}$ and $(-1)^j \varphi_\theta^{(j)}(x)$ being non-decreasing and convex on [0,∞), for x > 0, is the class of Archimedean copula. The function φ(·) is called the generator of the copula and commonly depends on a single parameter θ.

Figure 2:



Fully and partially nested Archimedean copula of dimension d = 4 with structures s=(((12)3)4) on the left and s = ((43)(12)) on the right.

At the moment, there are three different ways to estimate HAC:

- Ordinary (full) ML estimation, also denoted by FML, which is based on the complete log-likelihood and hence on a predetermined structure.
- The ML setup is based on realized pseudo-variables, i.e., the values of the pseudo variables for the given sample are explicitly computed, so that the bivariate density is maximized with respect to the copula parameter at each step of the procedure. The benefits of the ML method hold in particular for binary and non-complex structures. If the structure of the HAC is, however, complex, the estimates around the initial node seem to be slightly biased. Note that this procedure is not supported by asymptotic theory.
- More precise results can be obtained by the recursive ML (RML) procedure discussed in Okhrin et al. (2013a). The difference between the ML method and the recursive ML procedure results from the maximized log-likelihood. While the bivariate log-likelihood is considered at each estimation step of the ML method, the log-likelihood of the recursive ML procedure corresponds at each estimation step to the full log-likelihood for the marginal HAC regarded at that step. Compared to the full ML approach, the log-likelihood is only optimized with respect to the parameter at the root node taken the estimated parameter(s) at lower hierarchical levels as given, so that the final HAC being a copula is ensured by shortening the feasible parameter interval from above. From this point of view, the computational challenge is to build the log-likelihood for the full ML estimation, which is almost solved by constructing the d-dimensional density.

DISCUSSION

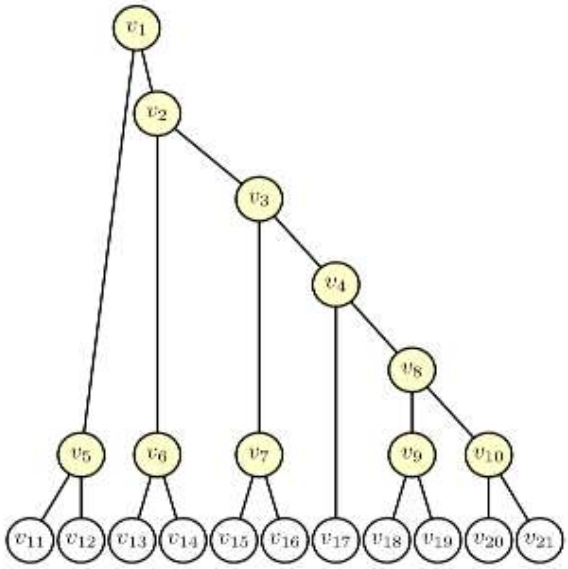
Existing System:

In existing system generalized quantifiers related to copulas are introduced. Fitting copulas to multidimensional data is a method for analyzing dependencies, and the proposed quantifiers of observational calculus assess the results of estimating the structure of joint distributions of continuous variables by means of hierarchical Archimedean copulas. Sufficient conditions for the function defining a hierarchical Archimedean copula to be indeed a copula, which have so far been rigorously established only for the special case of fully nested Archimedean copulas, hold in general. The paper concludes by comparing the proposed quantifiers to a more

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

traditional approach maximum weight spanning trees. In this system finding of nodes takes place in a worst case and found that time delay occurs due to long traverse in finding the required node.

Figure 3:



Tree structure (v, ε)

Table 1: Definition for the children and Labeling of Forks

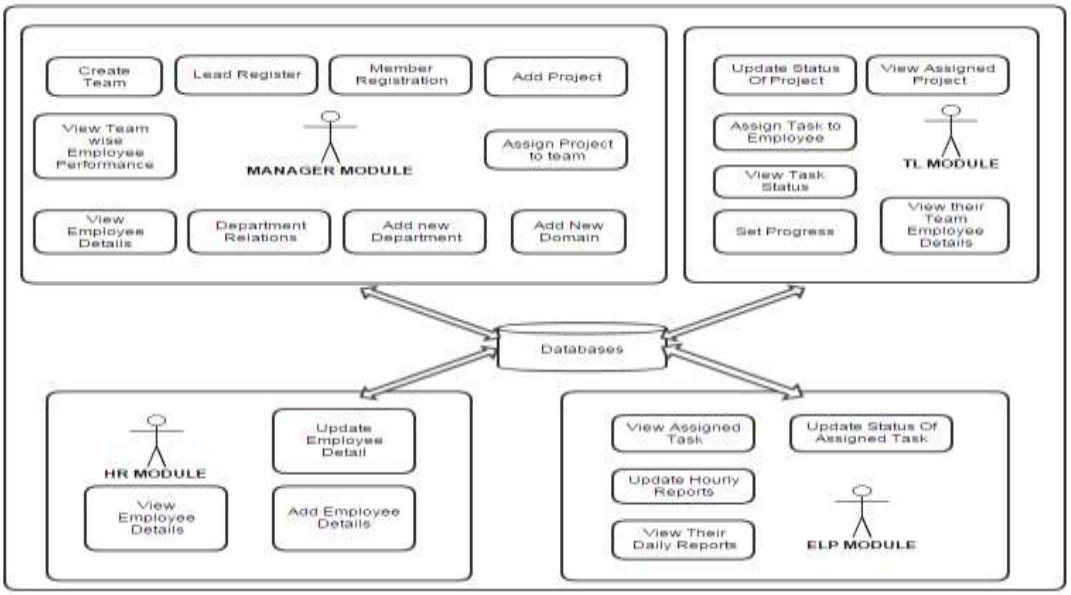
fork v	Children $\in \Lambda(v)$	Labelling $\lambda(v)$
v_1	v_2, v_5	$\Psi(.,1)$
v_2	v_3, v_6	$\Psi(.,1)$
v_3	v_4, v_7	$\Psi(.,1.1)$
v_4	v_8, v_{17}	$\Psi(.,1.2)$
v_5	v_{11}, v_{12}	$\Psi(.,1.1)$
v_6	v_{13}, v_{14}	$\Psi(.,1.1)$
v_7	v_{15}, v_{16}	$\Psi(.,1.3)$
v_8	v_9, v_{10}	$\Psi(.,1.4)$
v_9	v_{18}, v_{19}	$\Psi(.,2.6)$
v_{10}	v_{20}, v_{21}	$\Psi(.,1.8)$

Proposed System:

In this project we have exhaustively used the concepts of Hierarchical Archimedean copulas (HAC) algorithm. Here we have using three generalized quantifiers HAC distance, HAC Kendall's, HAC Distance + Kendall's. HAC distance quantifier is used to find the shortest distance to reach the required nodes.HAC Kendall's quantifier is to find threshold for the value of Kendall's rank correlation coefficient. Here we are using minimum spanning tree approach which helps to find the data rapidly even in a best case itself ,which means there will be no need long traverse and also less time consumption. Therefore, reduces time delay, no need of long traverse to find the nodes etc., Each time a step of the algorithm is performed, one edge is examined. If there are only a finite number of edges in the graph, the algorithm must halt after a finite number of steps. Thus, the time complexity of this algorithm is clearly O (n), where n is the number of edges in the graph.

Figure 3:

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785



System Architecture

Modules:
Modules have separated based on the roles which have been used in the application.

HR Responsibilities:
HR is the responsible person for adding of employees in this application. HR can add employee to the database and then updating or editing the employee details. HR can view all Employee details then HR can view the employee performance in the dashboard.

Technical Manager Responsibilities:
In this module is fully for Team Manager. All the works are done here by Team Manager only. Overall works and algorithm was implemented in this module by inserting and fetching of data's from databases. **HAC algorithm** is used in this module. From this module can able to create new team. After that we can register the Team Leader for created teams then add employees to that team. In this module can add new domain and department. After adding department and domain we give relation to them via ids then add new project to database and assign project to team. They can able to view the project list and assigned project working status and can able to view team member's performance by Bar and Line chart. They are also can view employee details. Manager can view employee details by domain and designation.

Leader Responsibilities:
In this module fully covers overall Team Leads responsibility. In this module TL can view the assigned project by Manager. She/he can able to assign the project to employee task by task or full project. Here can able to view the task completion status and then TL can update the status of the project which is assigned by Manager. TL can set progress for each and every employee who belongs to their team. Progress has been set based on the hourly reports send by ELP's. Here TL cans able view the progress report by Bar chart and Line chart. TL can view their Team member's basic details by this application.

Trainers Responsibilities:
ELP can view the tasks assigned by the team leader via notification. They can update the hourly reports and daily reports directly to the team leader and also they can report about their project status through this application.

CONCLUSION
The paper is directed towards the research into data mining and exhaustively uses the concepts of Hierarchical Archimedean copulas (HAC) algorithmthat aims at generalized quantifiers. It has proposed three generalized quantifiersHAC distance, HAC Kendall's, HAC Distance + Kendall's related to estimating the structure of joint

distributions of continuous variables by means of Hierarchical Archimedean copulas. HAC distance quantifier is used to find the shortest distance to reach the required nodes and HAC Kendall's quantifier is to find threshold for the value of Kendall's rank correlation coefficient.First proven are conditions that hold for HACs in general. Minimum weight spanning tree approach helps to find the data rapidly even in a best case itself; there will be no need of long traverse and also less time consumption. Spanning tree is very important in designing efficient routing algorithms. The proposed quantifiers are the first generalized quantifiers directly applicable to data that are realizations of continuous variables, without any preceding discretization. From data mining point of view, a deficiency of the existing theoretical fundamentals of HACs is that sufficient conditions for the function defining a HAC to be indeed a copula have been established only for fully Nested Archimedean copulas. In future, the FNAC algorithm techniques can be used and the dataset can be analysed or monitored.From our opinion, this shows that the agreement with data is positively influenced by the specific advantage of copulas.

ACKNOWLEDGEMENT

We would like to express our gratitude and greatest appreciation towards Assistant Professor N.DEEPA for giving us an opportunity to work under her guidance.

REFERENCES

[1] Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson, 2013.
[2] Data Mining: Practical Machine Learning Tools and Techniques (The Morgan Kaufmann Series in Data Management Systems), Ian H. Witten, Morgan Kaufmann, 2011.
[3] Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems), Jiawei Han, Morgan Kaufmann, 2011.
[4] Data Mining Techniques: For Marketing, Sales, and Customer Support, Michael J. A. Berry, Gordon S. Linoff, Wiley, 1997.
[5] Okhrin O, Okhrin Y, Schmid W (2013b). "Properties of Hierarchical Archimedean Copulas."
[6] Savu C, Trede M (2010). "Hierarchies of Archimedean Copulas."
[7] Whelan N (2004). "Sampling from Archimedean Copulas."
[8] J. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem."
[9] M. Holena and M.Scavnicky, "Application of copulas to data mining based on observational logic," in Proc. Inf. Techonol. Appl. Theory Workshops, Posters, Tuts., 2013.
[10] M. Delgado, M. Ruiz, and D.Sanchez,"Mining exception rules," in Proc. Found. Reasoning Uncertainty, 2010.

AUTHOR PROFILE:

V. VASANTHI
COMPUTER SCIENCE AND TECHNOLOGY,
G.K.M. COLLEGE OF ENGINEERING AND TECHNOLOGY.
CHENNAI, INDIA.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

	**G.B.YOGAVARSHINI** COMPUTER SCIENCE AND TECHNOLOGY, G.K.M. COLLEGE OF ENGINEERING AND TECHNOLOGY. CHENNAI, INDIA.
	**N. DEEPA** ASSISTANT PROFESSOR COMPUTER SCIENCE AND TECHNOLOGY, G.K.M. COLLEGE OF ENGINEERING AND TECHNOLOGY. CHENNAI, INDIA.