RESEARCH ARTICLE OPEN ACCESS

A Novel Approach of Clustering Using COBWEB

V.Kanageswari [1], Dr.A.Pethalakshmi [2]
Department of Computer Science
M.V.Muthiah Government Arts College for Women, Dindigul
Tamil Nadu - India

ABSTRACT
COBWEB Algorithm is a Hierarchical clustering algorithm. In this paper, the COBWEB algorithm constructs a classification tree incrementally by inserting the objects into the classification tree one by one. When inserting an object into the classification tree, the COBWEB algorithm traverses the tree top-down starting from the root node and find the best position to insert a new object by calculating the Category utility(CU) function. Here the best position of an object is find out according to the maximum value of the category utility function. At each node, the COBWEB algorithm considers four possible operations and selects one that yields the highest Category Utility function value. The four operations are create, Insert, Merge, Split. Create is the first step in Cobweb Algorithm. The node corresponding to the particular object should be created and it should be inserted into the Classification tree. When inserting the next node, we have to find out whether merging the corresponding node to the previous node is best or keeping the corresponding node as a separate split is best according to the maximum of CU function value. If the CU function Value of merging process is maximum then we have to merge currently selected node to the previously presented node of the classification tree else we have to keep the currently selected node as separate split of the classification tree. Among the four basic operations, the corresponding operation is selected based on the CU function value. Thus the COBWEB Algorithm incrementally organizes records into a tree.

Keywords:- Cobweb Model, Category Utility, Digital Object Identifier, Probability Distribution Functions.

I. INTRODUCTION

Data Mining

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge driven decisions.

Working Principles of Data Mining

While large scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data

based on open ended user queries. Several types of analytical software are available: Statistical, machine learning, and neural networks. Generally, any of four types of relationships are sought:

- **Classes:** Stored data is used to locate data in predetermined groups. For , a Restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.
- **Clusters:** Data items are grouped according to logical relationships or consumer preferences. For example, data can be mine to identify market segments or consumer affinities.
- **Associations:** Data can be mined to identify associations. The beer-diaper example is an example of associative mining.

- **Sequential Patterns:** Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likehood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

Clustering

Cluster is a collection of data objects which are similar to one another within the same cluster and dissimilar to the objects in other clusters. Cluster analysis means finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters.

Major Approaches

- **Cluster/2**: Flat, conceptual, batch learning, possibly overlapping, deterministic.
- **Partitioning methods**: flat, batch learning, exclusive, deterministic or probabilistic. **Algorithms:** k-means, Probability-based clustering.
- **Hierarchical clustering**:

Partitioning: Agglomerative (bottom-up) or divisible (top-down)

Conceptual: Cobweb, category utility function

Hierarchical Conceptual Clustering Cobweb

Incremental clustering algorithm builds taxonomy of clusters without having a predefined number of clusters. The clusters are represented probabilistically by conditional probability P(A=v/C) with which attribute A has value v, given that the instance belongs to class C. The algorithm starts with an empty root node and the instances are added one by one. For each instance the following options (operators) are considered:

- Classifying the instance into an existing class;
- Creating a new classes and placing the instance into it;
- Combining two classes into a single class(merging) and placing the new instance in the resulting hierarchy;
- Dividing a class into two classes (splitting) and placing the new instance in the resulting hierarchy.

COBWEB is an incremental system for hierarchical conceptual clustering. COBWEB incrementally organizes observations into a classification tree. Each node in a classification tree represents a class (concept) and is labeled by a probabilistic concept that summarizes the attribute value distributions of objects classified under the node.

The classification tree can be used to predict missing attributes or the class of a new object. Cobweb employs in building the classification tree. Which operation is selected depends on the category utility of the classification achieved by applying it. There are four basic operation and they are:

- **Merging two Nodes** - Merging two nodes means replacing them by a node whose children is the union of the original nodes sets of children and which summarizes the attribute value distributions of all objects classified under them.
- **Splitting a node** - A node is split by replacing it with its children.
- **Inserting a new node** - A node is created corresponding to the object and being inserted into the tree.
- **Passing an object down the hierarchy** - Effectively calling the COBWEB algorithm on the object and the sub tree rooted in the node.

COBWEB AS AN INCREMENTAL SYSTEM:

COBWEB is an incremental conceptual clustering system that can be viewed as hill climbing through a space of classification trees. The program does not adopt a purely agglomerative or divisive approach, but uses divisive (splitting) as well as agglomerative (merging) operators in tree construction. Schlimmer and Fisher (1986) propose three criteria for evaluating incremental systems that were inspired by Simon (1969). These criteria (adapted for conceptual clustering) are:

- The cost of incorporating a single instance into a classification tree.

- The quality of learned classification trees.
- The number of objects required to converge on a stable classification tree.

Generally in incremental systems, incorporation cost should be low, thus allowing real time update. However, this may come at the cost of learning lower quality classifications and/or requiring a larger sample of objects to find a good classification than a similarly based non-incremental, search intensive method. This section evaluates COBWEB using these criteria and verifies it to be an economical and robust learner. In this paper, Several clustering algorithms should be learned and reviewed the conceptual hierarchical clustering method in detailed and implemented in various datasets.

II. RELATED WORKS

Data clustering is a thrust area of research for statisticians as well as data mining researchers which resulted in the development of a vast variety of successful clustering algorithms. The COBWEB Algorithm was developed by machine learning researchers in the 1980s for clustering objects in an object attribute dataset. The COBWEB Algorithm yields a clustering dendrogram called classification tree that characterizes each cluster with a probabilistic description. The Cobweb algorithm operates based on the Category Utility function that measures Clustering Quality. If we partition a set of objects into m clusters, then the CU of this particular partition is given by using the equation.

There are several researches and implementations have been done on COBWEB model. In this paper, we reviewed some of the research papers on clustering algorithm and mainly a Cobweb model which is a conceptual hierarchical clustering algorithm.

Carl chiarella et al.[2] introduced a fairly non-linear supply function into the traditional Cobweb model under adaptive expectations. They found that the dynamics of the model were driven by a single hump map o the type that occurred in the chaos literature. By applying some recent results of feigenbaumthey were able to show that on its locally unstable region the Cobweb model exhibited a regime of period doubling followed by a chaotic regime.

Douglas H. Fisher et al. proved[3] that Conceptual clustering was a machine learning for unsupervised classification developed mainly during the 1980s. This paper presented COBWEB as a conceptual clustering system that organized data so as to maximize inference ability. Additionally, Cobweb was incrementally and computationally economical, and thus could be flexibly applied in a variety of domains.

Douglas H. Fisher et al.[4] described Cobweb algorithm that it was incrementally organized observations into a classification tree. Each node in a classification tree represented a class (concept) and labeled by a probabilistic concept that summarized the attribute value distributions of objects classified under the node. That classification tree could be used to predict missing attributes or the class of a new object.

A.Ketterlin et al. define [5] that many machine–learning (either supervised or unsupervised) techniques assumed that data presented themselves in an attribute-value form. This paper described that a clustering system was able to discover useful groupings in structured databases. It was based on the COBWEB algorithm, to which that added the ability to cluster structured object.

Mordecai Ezekiel et al. described[6] the mechanism of those self-perpetuating commodity cycles were well developed a decade or more ago, but despite various partial explanations, a definite theoretical explanation for them had not been established. Finally three economists, in Italy, Holland, and the United States, apparently independently, worked out the theoretical explanation which had been come to be known as the “COBWEB THEOREM”.

Ronald H.Coase et al. explained[7] the cobweb model’s irregular fluctuations on prices and quantities that may appeared on some markets. The

key issue on those models was time, and the way on which expectations of prices adapt determined the fluctuations in prices and quantities. That was on Kaldor's paper on the subject, "A Classificatory Note on the Determinateness of Equilibrium",1934,where the analysis of those models became of great interest,and where phenomenon took the name of cobweb theorem.

Tsang,S. Et al. Extended[9] traditional decision tree classifier to handle data with uncertain information, which originated from measurement/quantization errors, data staleness, multiple repeated measurements, etc.. They extended classical decision tree building algorithms to handle data tuples with uncertain values. Since processing Probability Distribution Function's was computationally more costly, they proposed a series of pruning techniques that can greatly improve the efficiency of the construction of decision trees.

III. METHODOLOGY

A. *Overview of cobweb algorithm*

Cobweb algorithm is a symbolic approach to category formation. It uses global quality metrics to determine number of clusters, depth of hierarchy, and category membership of new instances and here the categories are probabilistic. Instead of category membership being defined as a set of feature values that must be matched by an object, COBWEB represents the probability with which each feature value is present. It is also an incremental algorithm. Any time a new category or modifying the hierarchy to accommodate it.

B. *The cobweb algorithm*

COBWEB (root, instance):
Input: A COBWEB root node, an instance to insert record
If root node (C_0) is not created then
Create root node (C_0) and
Insert instance 'a' and 'b' as a separate cluster C_1 and C_2 into the root node (C_0)
Else if root node (C_0) is created then
While inserting next node
Assume the Insertion of the next instance in all possible positions
Calculate Category Utility (CU) fir each possible position of the clusters
If CU of merge > CU of split then
Merge (root node, instance)
Else
Split (root node, instance)
End if
Until all instances are inserted into the clustering tree
End Loop
End If

C. *The Category Utility Function*

The COBWEB algorithm based on the so-called category utility function (CU) that measures clustering quality. Category Utility attempts to maximize both the probability that two objects in the same category have values in common and the probability that objects in different categories will have different property values. If we partition a set of objects into m clusters, then the CU of this particular partition

$$\frac{\sum_{k=1}^{m} P(C_k)\left[\sum_i\sum_j P(A_i = V_{ij}|C_K)^2 - \sum_i\sum_j P(A_i = V_{ij})^2\right]}{m}$$

For a given object in cluster C_k, we guess its attribute values according to the probabilities of occurring, then the expected number of attribute values that we can correctly guess is

$$\sum_i\sum_j P(A_i = V_{ij}|C_K)^2$$

Given an object without knowing the cluster that the object is in, if we guess its attribute values that we can correctly guess is

$$\sum_i\sum_j P(A_i = V_{ij})^2$$

$P(C_k)$ is incorporated in the CU function give paper weighting to each cluster. Finally m is

placed in denominator to prevent over fitting. In cobweb algorithm the assumption that the attributes are independent of each other is often too strong because correlation may exist and is not suitable for clustering large database data and expensive probability distributions.

D. Implementation of cobweb algorithm

The Cobweb Algorithm was developed for both numerical and categorical attributes. The different datasets are collected from UCI machine learning repository[1] and applied Cobweb algorithm and is presented in Table 1.

Animal Data set Description

The Animal Data set contains four Attributes. All the attributes are both numerical and categorical. All the attributes are considered as input attributes. The Animal Data set provides the details about the features of animals and which could be useful to cluster the animals according to its features. Here the clustering should be done based on the category utility function which is a probabilistic measure of each data value in the table .

TABLE 1.ANIMAL DATA SET

INSTANCE LABEL	COLOUR	NUCLEI	TAILS
a	White	1	1
b	White	2	2
c	Black	2	2
d	Black	3	1

Solution:

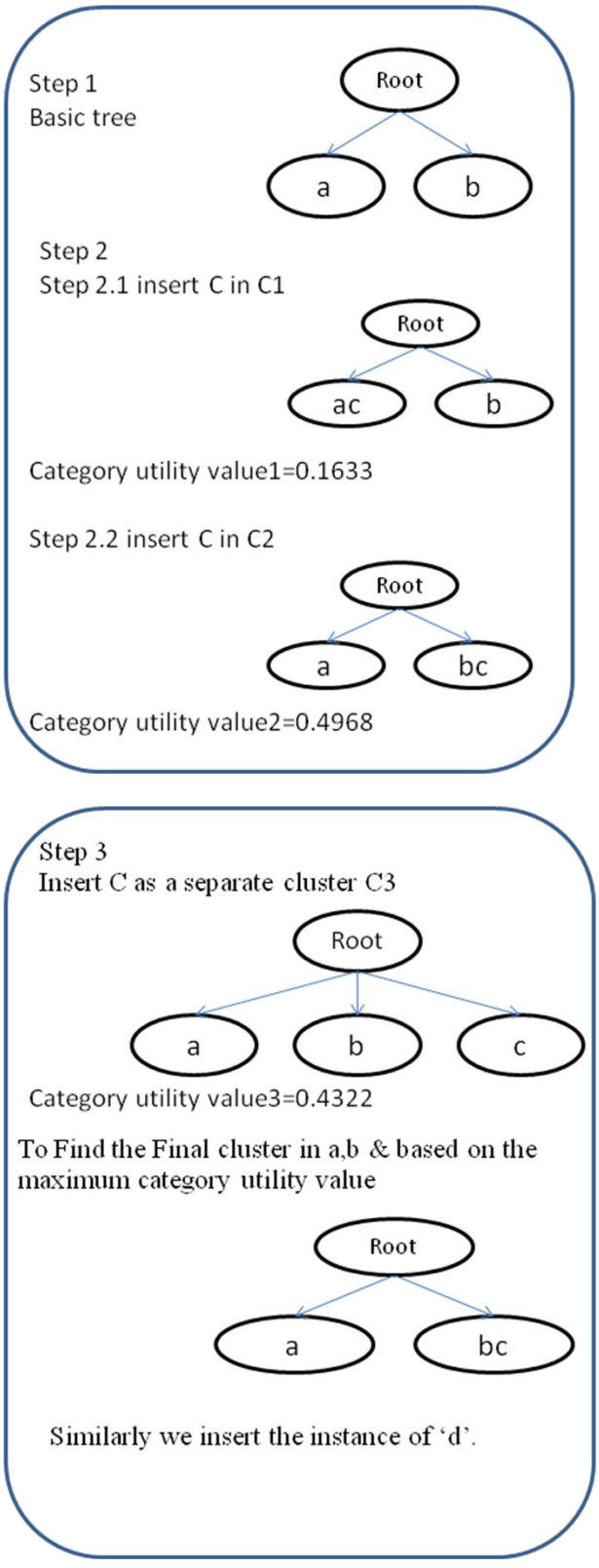

Figure 1: steps for implementation

TABLE 2.UCI DATASET USED IN COBWEB ALGORITHM

S.NO	UCI Data set	Attributes	Final clusters
1	Animal Data Set	Instance Label: Colour, Nuclei, Tail	a, bc, d
2	Banana Data Set	Instance Label: Size, smell, Colour, Field	a, b, c, df, e, gh
3	Car Data Set	Instance Label: Weight, Door, Size, Cylinder	ad, bgh, cef

Above the table produce final clusters for the datasets. **A**nimal dataset produce three clusters using four attributes. Banana dataset produce six clusters using five attributes. Car dataset produce three clusters using five attributes.

IV. CONCLUSION

In this paper, we designed and developed a COBWEB algorithm that was generalized to read any categorical as well as numerical data set with any number of attributes. In this COBWEB algorithm, a heuristic measure such as Category Utility Function was used to calculate the probability of each data value and which could be helpful in clustering the instances according to maximum value of the Category Utility Function. The COBWEB algorithm was tested on different data sets from the UCI machine learning repository. I observed that the Cobweb algorithm find out the best clusters based on the value of the data objects.

REFERENCES

[1] C. L. Blake and C.J. Merz, "UCI Repository of Machine Learning Databases", Irvin, University of California, http://www.ics.uci.edu/~mlearn/,1998.

[2] Chaos Carl Chiarella "The Cobweb model", Article In Scientific American Journal, 1987.

[3] H. Douglas Fisher, "Proceeding of the AAAI Conference" Seattle Washington. Page 461-465, 1987.

[4] H. Douglas Fisher, et al, "Knowledge acquisition via incremental conceptual clustering". Machine learning2(2):s139-172.(DOI):10.1007/BF00114265,1987.

[5] A.Ketterlin, P.Ganasarski& J.J. Korczak LSIIT, " Conceptual Clustering in Structured Databases: A Practical Approach" AAAI Conference,1995.

[6] Mordecao Ezekiel, S.Benner, Nicholas Kaldor et al. "The Cobweb Theorem" The Quarterly Journal of Economics, Volume52, Page255-280.

[7] Ronold H.Coase, Wassily Leontief or NicholasKaldor. "Cobweb model". Journal of Policonomics 2012.

[8] Rina Dechter, Judea Pearl,"Tree clustering for constraint networks" Artificial Intelligence, Volume 38,Issue 3 ,Pages 353-366, April 1989.

[9] Tsang S;Kao,B.;Yip,K.Y; Wai-Shing Ho"Decision Trees for Uncertain Data",Proceedings of Data Engineering, ICDE'09.IEEE 25th International Conference ,Pages 441-444,2009.

[10] Wikipedia - "Conceptual Clustering", http://en.wikipedia.org/wiki/Conceptual_clustering,https://fenix.tecnico.ulisbo.pt/downloadfile/3779574227995/licao_13.pdf

[11] William Sia,Mihai M.Lazarescu,"Clusterig Large Dynamic Datasets Using Exemplar Points", Machine Learning and Data Mining in Pattern Recognition,Page163-173,2005.

[12] Ying Zhao and George Karypis, "Evaluation of Hierarchical Clustering Algorithms For Document Datasets", Proceedings of Eleventh International conference on Information and knowledge Management,Page 515-524,2002.

.