A Dynamic GSOM-based Concept Tree for Capturing Incremental Patterns

Pin Huang, Susan Bedingfield
Faculty of Information Technology
Monash University
Melbourne, Australia
e-mail: phua13@student.monash.edu
e-mail: sue.bedingfield@monash.edu

Damminda Alahakoon
Latrobe Business School
Latrobe University
Melbourne, Australia
e-mail: d.alahakoon@latrobe.edu.au

***Abstract*—The Growing Self Organizing Map (GSOM) has been proposed to address the need of predefining network size and shape in traditional Self Organizing Maps (SOM). In the work described in this paper, the GSOM is used as a foundation for generating hierarchies of concepts in a tree structure which also has the ability to adapt and accumulate new information in an incremental learning architecture. GSOMs are used to capture inputs in time windows and the GSOM nodes are used as the base for developing the bottom level concepts in the tree. A new algorithm is then used to integrate similar information into concepts based on attribute similarities. As new data is introduced, new GSOMs are created and used to capture topological patterns which are integrated into the existing concept tree incrementally. The updated concept tree can capture multiple dimensional inputs with multi-parent nodes. It is proposed that this is an ideal building block to implement the columnar architecture in the human neo-cortex as an artificial model which could then be used as a cognitive architecture for data mining and analysis. The adaptive concept tree model is demonstrated with several benchmark data sets.**

Keywords-growing self organizing map; clustering; concept formation; incremental learning.

I. INTRODUCTION

According to current brain theories, human intelligence and related factors, such as perception, language, prediction, all have a strong relationship to the architecture and structure of the neocortex. The neocortex is believed to be a complex biological auto-associative memory [5], where one of the key features is that patterns from 'experiences' (inputs) are stored in the neocortex in the form of a hierarchy [5]. When storing these patterns, the cortical region provides the group of related active cells a name, and this name is passed to the next higher level in the hierarchy; only the representation of the active cells is passed via the hierarchy; and when the patterns move down the hierarchy, the higher level concepts are broken into granular information [5]. The work described in this paper is based on this base functionality and structure of the neocortex resulting in a model which can capture and accumulate patterns from input data and also adapt to changes with incremental learning. In our proposed concept tree model, lower level represents a more detailed concept and higher level is about a more abstract concept. The information passed from a node at a lower level to a higher level of the tree consist of a median weight value and as such only abstract representative information and no detailed actual information is passed up the hierarchy. This ensures that only high level concepts are captured in the upper levels of the hierarchy.

A further key feature of the neo cortex is that patterns are stored in sequence and activated in sequence with appropriate triggering mechanisms [5]. When we recall our memories, we have to go through it in a sequential order. Although the current version of the proposed model does not demonstrate this functionality, the dynamic and adaptive architecture of the proposed model is an ideal base for developing such capability. This work is currently ongoing as the second phase of the project.

Mountcastle [13] believed that the structure of the neocortex has a columnar organization. The term column can be viewed as a vertical unit in which cells work together. And such columnar unit is the basic computation unit for the cortical computation. The proposed concept tree model is an incremental learning model, which is capable of continuously processing incoming data and adapting as required. The model has the capabilities of generating new columns of sub columns when the new data do not exactly represent past happenings.

The proposed model provides a basis for a larger artificial learning and adaptive model being planned, which can capture accumulate and represent data in a form suitable for decision making. The proposed model is inspired by the current research findings of the neocortex and columnar structure of the brain; therefore, the proposed model embraces some key features: hierarchical concepts formation, incremental learning and adaptation, columnar structure. The GSOM-based tree structure presented in this paper will form an individual column in the larger model with each sub child column representing sub groupings and concepts within each column.

The proposed architecture is made up of three key components: GSOM clustering generated input, a tree base hierarchy, and an incremental update mechanism to accommodate new inputs. Section 2 provides the background for the work described in the paper. The new model and architecture is described in detail in section 3. Experimental results with two benchmark data sets are described in Section 4. Section 5 provides concluding remarks.

 ISBN: 978-1-61208-390-2

II. BACKGROUND

Mountcastle [13] proposed that the structure and appearance of the neocortex is quite uniform and comprises columnar units that run perpendicular to the horizontal layers of the neocortex [13]. The term *column* can be viewed as a vertical unit in which cells work together. Such a columnar unit is the basic computation unit for the cortex's operation. The human neocortex is described as being composed of several hundred millions of mini-columns. Mountcastle [13] also suggested that a cortical area may belong to more than one column or sub column. In other words, a cortical area located in a lower hierarchical level may relate to more than one cortical areas in higher hierarchical levels. This biological feature enables us to relate experiences or inputs to multiple concepts. To accommodate such capability our proposed model enables a child node of a lower level to have more than one parent node of higher levels.

Hawkins [5] has also suggested some key features of the neocortex. For example, patterns are stored in the neocotex in sequence and in the form of hierarchy. Based on his theory of the neocortex, Jeff Hawkins has proposed a Hierarchical Temporal Memory (HTM) model to capture such functionality based on Markov chains [10] and Bayesian belief propagation. These techniques are considered to be symbolic techniques (which deal in human defined abstract symbols) and it has been discussed by Weng [6] that emergent techniques (which can autonomously self-organize via past experience) are more suited to achieving similar functionality to the neocortex. Emergent models include the self-organizing techniques. Our proposed model is based on the GSOM [1].

The GSOM is an unsupervised neural network and has the ability to grow dynamically, the necessity for overcoming the major limitation of the SOM algorithm of a predefined map size. The GSOM algorithm facilitates hierarchical clustering using the Spread Factor (SF) parameter. With a lower SF, a more abstract map can be obtained whereas with a higher SF, a more detailed map can be obtained. In our proposed model, we use a high SF to obtain a very detailed map, which is the building block for the construction of the concept tree. Each node produced from GSOM is viewed as a mini or sub column. In addition, each concept tree which is composed of several hierarchical levels generated from the proposed model, can be viewed as a columnar unit, and its sub trees can be viewed as sub columns. Earlier conceptual clustering models such as CLUSTER/2 [11], do not have incremental learning capability, in contrast, the learning of the human process of incremental knowledge acquisition. There are some incremental conceptual clustering models such as EPAM [3], UNIMEM [9], COBWEB [2], CLASSIT [8], which use different approaches to construct concept trees, however, they do not enable a child concept node to have more than one parent concept node, which means that the model cannot fully implement the neocortex hierarchical structure in which a child node in a column may have more than one parent node located in more than one column.

Lastly, incremental learning related to cognition has been described by Chalup [12] as the development of the brain functionality in three phases. Phase one is the incremental learning that occurs as a result of the evolutionary process over generations. Phase two refers to the neurodevelopment of the brain. This is the stage of acquiring essential abilities such as sensory perception and cognition. Phase three is about the adaptation of the neural system subject to the brain's internal state and the interaction with the environment. Therefore, one of the key features of the proposed algorithm is incremental learning.

III. ADAPTIVE CONCEPT TREE MODEL

A. GSOM

The GSOM algorithm has two modes, the training mode and testing mode. Actual growth of the network and smoothing out of weights occur in the training mode. In the testing phase final calibration of the network occurs if known inputs are used, and for unknown inputs the distance from the existing clusters in the network can be measured.
The training mode consists of three phases. Processing in those three phases is as follows [1].

1) Initializing Phase

a) Weight vectors for the starting nodes are initialized to random numbers between 0 and 1. In general, each map starts with four nodes.

b) Growth Threshold (GT) is calculated for the given data set based on user requirements. To calculate the GT, the SF parameter value, which is defined prior to the clustering, is used. The formula is $GT = -D * \ln(SF)$; *here D is the dimension of the input.*

2) Growing Phase

a) Input is presented to the network.

b) The weight vector closest to the input vector is selected using a similarity measuring function. The closest node is considered to be the winner node. The weight vector adaptation takes place for the winner node and the neighbourhood nodes. The amount of adaptation is based on the Learning rate (LR) parameter which is decreased exponentially over the iterations.

c) The error value of the winner node is accumulated by the difference between the winner node's weight vector and the weight vector of the input node.

d) If $TE_i > GT$, *where* TE_i *is the total error value of node* i *and GT is the Growth Threshold, then new nodes are inserted into the map if node* i *is a boundary node. If node* i *is a non-boundary node, the error value is distributed to the neighbourhood nodes.*

e) If new nodes are added, weight vectors are initialized to match the neighbouring node weights and initialize the learning rate to the starting value.

f) Repeat the above steps until all inputs are presented to the network and the node growth is set to a minimum level.

3) Smoothing Phase

a) Reduce the learning rate and define a small starting neighbourhood.

b) Present input weight vectors then find winners and adapt their weight vectors and the weight vectors of the neighbourhood nodes in a similar way to the growing phase.

The GSOM algorithm facilitates hierarchical clustering using the SF parameter. SF parameter value is used for the GT calculation and when the SF value is low the GT becomes high, making new node insertion more difficult. In contrast, when the SF value is high the GT becomes low, making new node insertion easier. Because of the above relationship the SF parameter value controls the growth of the output map. Using a lower SF value a more abstract map can be obtained whereas using a higher SF value, a more detailed map can be obtained. This functionality can be used for hierarchical clustering of a given dataset by obtaining an abstract map for the first level of the hierarchy and then further explore the map using a higher SF value.

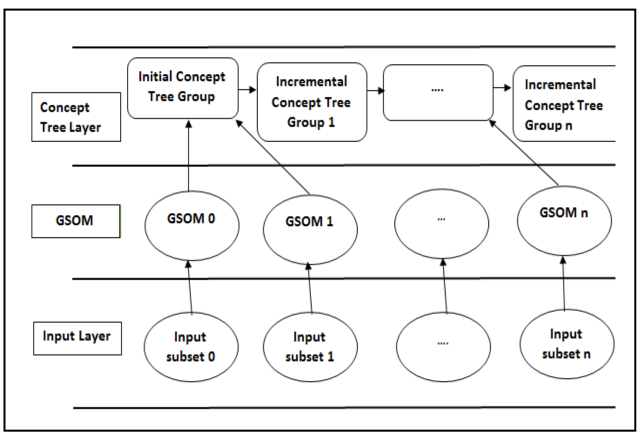

Figure 1. Overall Architecture

B. Overall architecture

The proposed architecture is made of three layers, namely, the input layer, the GSOM layer, and the concept tree layer. This is illustrated in Figure 1. The input layer is where the input data is located. The input dataset can be randomly broken down into several sub datasets. If the input dataset contains temporal features, the input dataset can be broken down by temporality, such that, sub datasets can be organized in a sequential order to represent such temporality. The number of sub datasets should be at least two. When the input dataset has been broken down, the sub datasets will be processed by the model in a sequential order. Furthermore, the number of GSOMs located in the GSOM layer is the same as the number of sub datasets in the input layer. Each GSOM in the GSOM layer is responsible for processing only one sub input dataset. When the first sub input dataset is presented to the first GSOM in the GSOM layer, the output of the GSOM will be presented to the Concept Tree Layer to form the initial concept tree group. After that, the second sub input dataset is presented to the second GSOM in GSOM layer, and then the outcome of the GSOM is presented to the previous established initial concept tree group to form the incremental concept tree group. Similarly, once the sub input dataset has been processed by its corresponding GSOM, the outcome of the GSOM will be presented to previous established concept tree group to generate the next incremental concept tree group.

C. GSOM Layer and Concept Tree Layer Architecture Details

After each GSOM is processed, it presents the clustering results to the bottom level of the previous existing concept tree group, which is illustrated in Figure 2. The concept tree group is composed of three level concept trees (noted as Tree 1 in Figure 2), two level concept trees (noted as Tree2 in Figure 2), and standalone nodes. A standalone node is a level 3 node that does not have any parent nodes at higher levels. Once the bottom level of the concept tree group has processed the input, the information will move up to higher levels. A higher level of the hierarchy means a more abstract concept than a lower level. We set the maximum number of the tree hierarchy to be three; however, the number of hierarchical levels can be set to be more than three by reapplying the same proposed methodology.

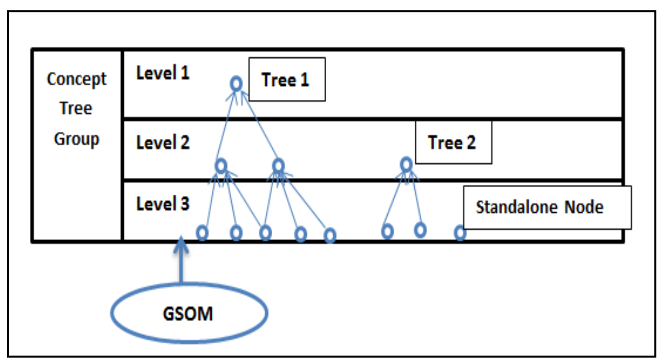

Figure 2. Concept Tree Group

D. Incremental Concept Tree Algorithm

1) Constructing the Initial Concept Tree

Inputs are first presented to the GSOM algorithm. If the value difference of a specific attribute for a pair of nodes agrees to within a predefined value, we say that they have *similar attribute values*. We set this predefined value to be 0.2, which is reasonable because the attribute values are between 0 and 1. Speak of which, attributes' values should be normalized before being presented to the algorithm. In addition, we say that two nodes *share the same concept* if a predefined percentage of their attribute values are similar. In this paper we use a value of 80% as the predefined percentage. For example, if there are two GSOM nodes (N1 and N2) with m attribute values. N1's attribute values are noted as (A1,A2,…,Am), N2's attributes values are noted as (B1,B2,….,Bm). If the absolute value of (Ai-Bi) is less than 0.2 (here i = 0,1,...,m), we say that N1 and N2 *share similar attribute values* for the ith attribute of the input data. If N1 and N2 share similar attribute values of more than m*20% attributes, we say N1 and N2 share the same concept. Information from the GSOM is first refined then transferred from the GSOM to level 3 (bottom) of the initial concept tree by successively merging the closest node pairs if they share the same concept.

a) Genertating Parent concepts at level 2 for similar nodes at level 3 of the initial concept tree group

For developing concepts from level 3 into level 2 (higher level), two level 3 nodes are defined to be similar if the Euclidean distance between the nodes is less than a predefined distance threshold. We set the threshold as 0.2 * square root of the number of attributes in the input data, which represents the maximum overall distance for all attributes. A parent node of these nodes will be generated at level 2. If a node cannot find any similar node to generate a concept at a higher level, this node will be a standalone node at this level.

b) Generating parent concepts at level 1 for similar nodes at level 2 of the initial concept tree

Similarity between nodes at level 2 is defined in the same way as at level 3. However, because level 1 parent nodes represent more abstract concepts than level 2 nodes, it is appropriate to use a wider distance threshold. We set the distance threshold as 0.4 * square root of the number of attributes of the input data. Parent nodes are created at level 1 for groups of similar nodes at level 2. If a node at level 2 cannot find any similar nodes to form a parent concept at level 1, this node will be without any parent nodes at level 1.

2) Incremental Learning Stage

When the next subset of the input data being presented to its corresponding GSOM, GSOM output nodes are further refined by grouping any closet nodes with similar concepts. Those nodes will be treated as a series of incoming input nodes to level 3 of the already existing concept tree group. If there is no node similar to the input node at level 3 of the existing tree group, the input node will be placed as a standalone node at level 3, which is illustrated in Figure 3.

If the Input node can find similar nodes at level 3, if there is no existing parent node at level 2 able to hold all the child nodes, a new parent node will be created at level 2, which is illustrated in Figure 4. What is more, this enables a child node to have more than one parent node in our proposed model. This new node at level 2 will be treated as an input node to the existing level 2. A similar mechanism of creating a parent node for level 3's nodes is applied to level 2 as well. If the most recently created node at level 2 cannot find any similar nodes at level 2, the node will be added to level 2 as another node. This update will continue up to level 1. Details are illustrated in Figure 5, which is the pseudo code of the incrementally adaptive concept tree algorithm.

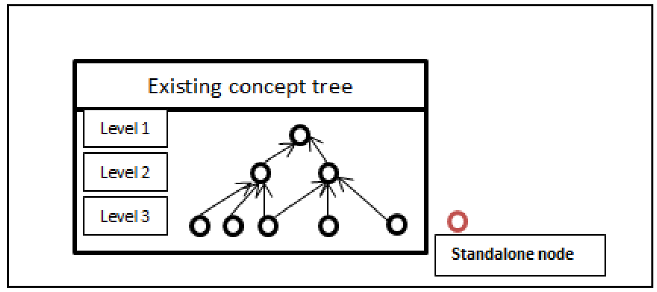

Figure 3. An example of a standalone node at level 3

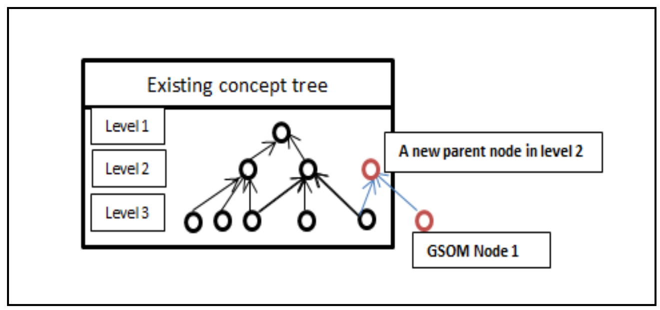

Figure 4. An example of generating a new parent node at level 2

IV. EXPERIMENTAL RESULTS

Experiments were run on two datasets (zoo dataset [7] and heart disease dataset [4]) from UCI data. The zoo dataset is composed of 17 attributes and 101 instances, a majority of attributes are of Boolean type. The Heart disease dataset's attributes are either continuous or Boolean type with 303 instances. The two data sets were chosen to demonstrate the functionality of the new algorithm. The zoo data has been widely used to demonstrate clusters and hierarchical clustering due to the availability of main animal groups and sub groups within. It is also interesting to have animals such as platypus and turtle etc. and see what the algorithm will do with such animals. The key advantage of using this data set is that we can understand why certain animals are grouped together from general knowledge. Also the animal data set has been used to demonstrate the clustering and hierarchical clustering ability of the GSOM and it was the ideal data to show how such clusters are used as a base for concept building and also the incremental update of such concepts. The heart disease data was selected as a more realistic data set, but with attributes which also has meaning to a general reader. As such it must be emphasized

```
Algorithm 2 Incrementally Adaptive Concept Tree
  When an input node is presented to level 3 of an existing concept tree
  Variables:
  DistanceThreshold = LevelFactor * square root of the number of attributes of the input data
  LevelFactor = 0.2 (level 3)
  LevelFactor = 0.4 (level 2)

  Procedure Adapting the concept tree in each hierarchical level (Node InputNode)
    Locate a winner node W by computing Euclidean distance.
    If the distance d between InputNode and W <= DistanceThreshold Then
       If W does not have a parent node
           Create a node P in a higher level as the parent node of the W and InputNode
           If node P is in level 2 Then
             Procedure adapting the concept tree in each hierarchical level (Node P)
           End If
       Else If W has parent nodes
           For Each ParentNode PP in W' Parent nodes
               If InputNode is similar to all PP's Child Nodes Then
                 Let InputNode be a child node of PP
               ELSE If InputNode is similar to a subset of PP's child nodes Then
                     Create a new node P in a higher level as the parent Node for them
                     If node P is in level 2 Then
                       Procedure adapting the concept tree in each hierarchical level (Node P)
                     End If
               End If
           End For
       End If
    Else
      Add the InputNode to level 3 of the existing tree
    End IF
  End Procedure
```

Figure 5. Incremental Concept Tree Algorithm

that the purpose at this stage is not to evaluate the accuracy of classification of the algorithm, but to demonstrate how GSOM based clusters are used as a base for multi-level concept building with incremental update. At this stage the 'meaningfulness' and 'explain ability' of the concepts are used to evaluate the algorithm. The GSOM has been fully evaluated for cluster accuracy, topology preservation capability and processing advantages. In the following experiments we demonstrate that such GSOM clusters can then be used to develop the concepts which could then be updated as new data changes without losing past learning.

For each node of different hierarchical levels, we calculate the nodes' weighted values and standard deviations for each attribute. These are used to identify the concepts in different hierarchical levels. With the zoo dataset, 16 attributes were used except the last attribute that indicates the animal's category. With the heart disease dataset, null values were removed. Fourteen attributes were used in the experiment, including age, sex and chest pain type, excluding "the diagnosis of the heart disease" attribute. Distinct values of the excluded attribute are 0,1,2,3 and 4, which indicate the probability of having heart disease. The value 0 means absence of heart disease (with less than 50% diameter narrowing), and the value 1,2,3 and 4 stands for different degrees of presence of heart disease (with more than 50% of diameter narrowing). We used a SF of 0.9 to run the GSOM for any subsets of dataset to obtain more detailed maps.

A. *Zoo dataset Results*

The dataset was divided into two subsets and input to two GSOMs separately. Five concept trees with three levels, six concept trees with two levels, and six standalone nodes were generated from the algorithm.

1) Three level hierachical Concept Tree

The input animals for each concept tree are shown in Figure 6. Tree 0 represent birds, tree 1 is a concept tree for mammals, and tree 2 represents fish. Trees 3, 4 and 5, they all represent reptiles and share some grandchildren (toad, slowworm, and newt).

Top level information provides a general idea about the most abstract concepts. The concept of a node is determined by each attribute's standard deviation and weight values. When an attribute's deviation value is 0, we say that all input instances attached to this node share the same concept, and such concept's name is the attribute's name and the actual value of the concept is determined by the weight values of the attribute. If a node has more than one attribute with zero standard deviation, the concept of a node is the collection of all these attribute's concept. For example, Figure 7 shows Tree 0's top node's attributes' weight values and standard deviations at level 1. The highlighted attributes with 0 standard deviations in Figure 7 stand for the concepts. Animals belonging to Tree 0 share the following concept at level 1: they do not have hair, have feathers, can produce eggs, do not have teeth, have backbones, can breathe, do not have fins, are not venomous, have tails, have two legs – as such birds.

Tree Name	Animals
Tree 0	crow, gull, hawk, kiwi, flamingo, duck, lark, chicken, skua, sparrow,swan,wren, dove, parakeet,pheasant,skimmer
Tree 1	boar,cheetah,leapard,lion,elephant, giraffe,gorilla,calf,lynx,antelope,buffalo, deer,goat,aardvark,bear,mole,opossum, squirrel,vole,mink,pony,pussycat,reindeer, mongoose,wallaby,seal,sealion,polecat, puma,racoon,wolf,oryx
Tree 2	catfish,chub,herring,seahorse,carp,haddock, dogfish,bass,sole
Tree 3	tortoise,toad,newt,slowworm,tuatara
Tree 4	newt, slowworm,tuatara,toad,scorpion
Tree 5	scorpion, newt,slowworm,tuatara,piviper

Figure 6. Three level Concept Tree's input animals

Child nodes inherit their parent nodes' concepts. This is shown in Figure 8. Input instances belonging to Node 3 at level 2 not only share concepts with their parent node at level 1, but also share the concept that animals are not domestic. Similarly, node 9 at level 3 inherits its parents' concepts, and input instances attached to this node also share two more concepts: being predator and not catsize (not the same size as a cat). Therefore, crow, gull, hawk and kiwi are predators and they are not the same size as a cat, they also have the concepts from parent nodes at level 1 and 2. Some nodes at level 3 have more than one parent at level 2 such as Node 7 at level 3. Node 7 and 13 at level 3 inherit the same concepts from their parent (Node 7 at level 2), but they differ in the concept of being domestic or not. Node 7 and Node 8 at level 3 have the same concept inherited from their parent node (Node 2 at level 2), but they differ in the concept of being aquatic or not.

Attribute	Standard Deviation	Weight Values	Attribute	Standard Deviation	Weight Values
hair	0.00	0.00	backbone	0.00	1.00
feathers	0.00	1.00	breathes	0.00	1.00
lay eggs	0.00	1.00	venomous	0.00	0.00
produce milk	0.00	0.00	fins	0.00	0.00
airborne	0.24	0.95	tails	0.00	1.00
aquatic	0.46	0.28	domestic	0.39	0.11
predator	0.48	0.18	catsize	0.33	0.20
toothed	0.00	0.00	legs	0.00	0.25

Figure 7. Tree 0's Standard Deviation and Weight Values

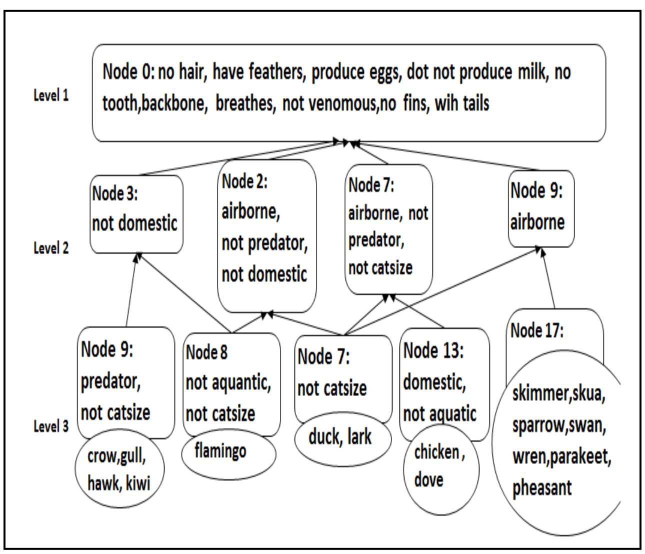

Figure 8. Three Level Concept tree

2) Two level concept trees

These are concept trees which could not be grouped with other nodes to form a more abstract concept at level 1. Figures 9 and 10 illustrate such trees related to aquatic creatures. When compared with the existing three level concept tree, Tree 2 in Figure 6, whose level 1 concept *is no hair, no feathers, produce eggs, no milk, not airborne, aquatic, toothed, backbone, do not breathe, not venomous, fins, tails, no leg*s. In Figure 9, octopus, seawasp are not toothed, and some animals in Figure 9 have legs; therefore, this is different from the concept of Tree 2 (no legs and toothed). Similarly, two concepts (milk and catsize) in Figure 10 are different from the concepts in Tree 2; therefore, trees in Figure 9 and 10 cannot be grouped with Tree 2. Figure 11 shows another category different from any concepts in Figure 6. Figure 12's tree shows two animals, cavy and hamster that do not produce milk, however, animals in Tree 1 do produce milk; therefore, they are under different concept trees.

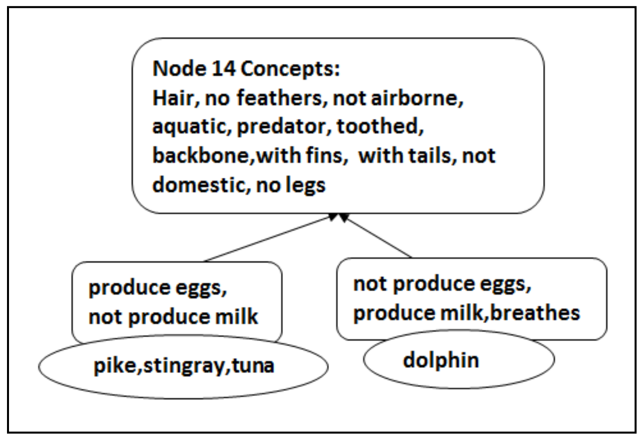

Figure 9. Two Level Concept trees with sea creature 1

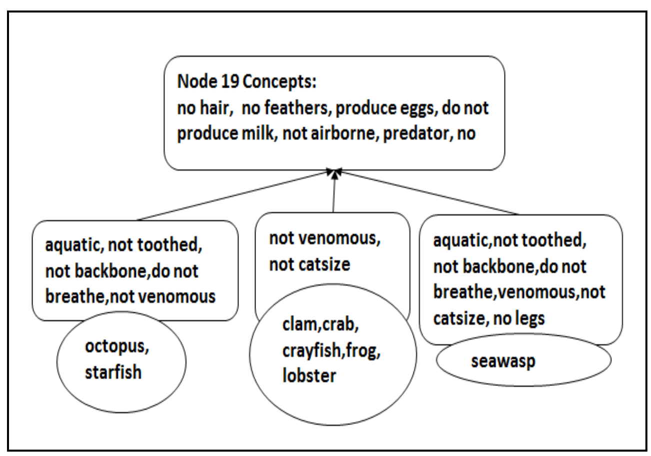

Figure 10. Two Level Concept trees with sea creature 2

3) Standalone nodes at level 3

Figure 13 shows standalone nodes at level.3, which are very different from other animals. Platypus has hair, which is

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-390-2

different from any aquatic animals having parent nodes at level 2 or 3. The seasnake does not produce milk or lay eggs, so it is a sea creature. The fruitbat is an airborne mammal, so it is differs from birds. The ostrich, penguin, rhea and vulture are all big birds. The slug, termite, and worm are not predators, not toothed, and do not have a backbone, therefore, reptiles shown in Figure 6 are different from them.

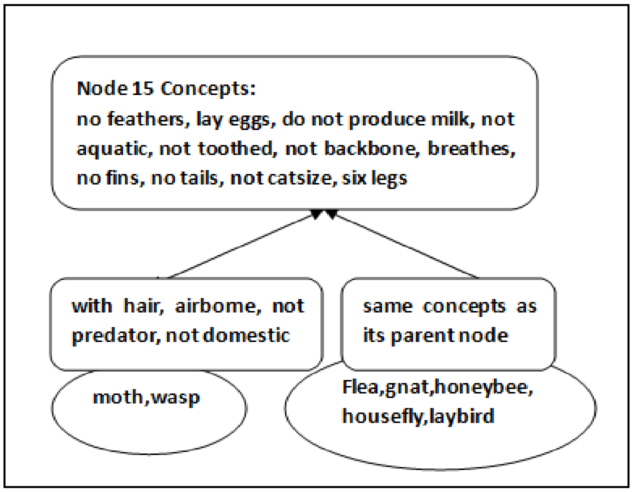

Figure 11. Two Level Concept trees with insects

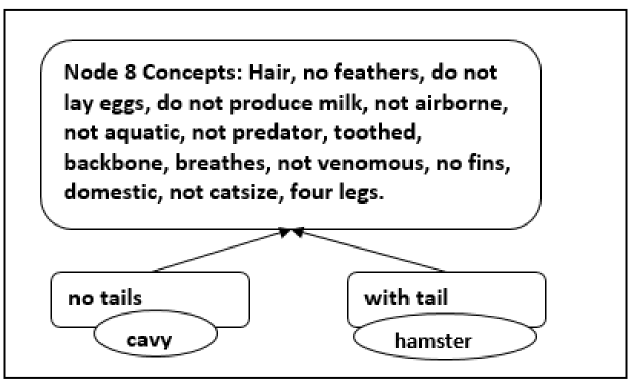

Figure 12. Two Level Concept Tree for cavy and hamster

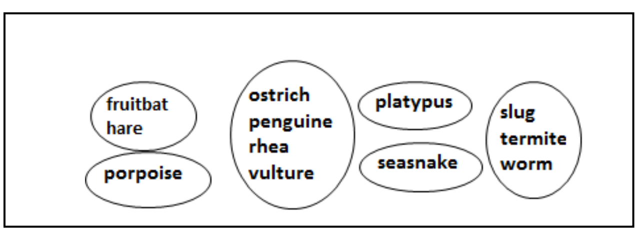

Figure 13. Level 3 standalone nodes

B. *Heart Disease dataset Results*

1) *Three level hierarchical trees*

Ten concept trees with three hierarchical levels were created. Figure 14 shows the first level concept in each tree and the percentage of instances belonging to each tree that do not have heart disease. When people do not have anginal pain, more than 80 % of instances under each tree do not have heart disease; when people suffer from anginal pain, it is very likely have heart disease (refer to 18.9% in Tree 0 and 10.4% in Tree 2). Therefore, anginal pain is a very important feature in the diagnosis of heart disease. When a person has anginal pain and "reversable defect", the probability of absence of heart disease increases if they do not have "graphic left hypertrophy". When we analyse concepts from Tree 4 and 5, we can conclude that if people have anginal pain but are "asymptomatic" and "normal (no defect)", their probability of having heart disease decreases compared with instances in Tree 0 and Tree 2. Tree 6, 7 and 8 have 100% of absence of heart disease, showing that when females do not have certain symptoms (indicated in each Tree), they will not have heart disease. From concepts indicated in Tree 6, 7 and 8, we notice that they share some common concepts, such as female, non-anginal pain.

Concept Tree	Tree 0	Tree 1	Tree 2	Tree 3	Tree 4
Level 1 Concept	not atypical angina, anginal pain, no graphic left hypertrophy, reversible defect	non-anginal pain, graphic normal	anginal pain, graphic left hypertrophy, reversible defect	non-anginal pain, graphic left hypertrophy	anginal pain, not asymptomatic, graphic left hypertrophy, Normal (no defect)
Percentage of Absence of Heart Disease	18.9%	81.0%	10.4%	82.1%	72.2%
Concept Tree	**Tree 5**	**Tree 6**	**Tree 7**	**Tree 8**	**Tree 9**
Level 1 Concept	female, asymptomatic, not graphic wave abnormality, not graphic left hypertrophy, normal	female, non-anginal pain, zero fasting blood sugar, not exercise induced angina, normal	female, non-anginal pain, not graphic wave abnormality, not graphic left hypertrophy, normal	female, non-anginal pain, not graphic left hypertrophy, not exercise induced angina, normal	asymptomatic, graphic left hypertrophy, normal
Percentage of Absence of Heart Disease	66.7%	100%	100%	100%	35.3%

Figure 14. Content from the three level concept tree

Figure 15 shows that Tree 6, 7 and 8 share some child nodes at level 2. Tree 9's level 1 concept indicates that people with the properties indicated in Figure 14 are more likely not to have heart disease, however, only about 35% of them do not have heart disease. The reason for this is explained by the concept tree as follows. Figure 16 illustrates details of concept tree 9, in which, 'No of Prob_0: 1' means the number of the instances with probability type (the degree of having heart disease) of 0 is one. Similarly, 'No of Prob_1:2' means the number of instances with the probability type of 1 is 2. Node 45 at level 3 has only one instance. Because of this, we only show concepts that are comparable to sibling nodes' concepts. Node 45 and 64 share the same concepts: zero fasting sugar and zero major vessel, but one group is male, the other group is female. Due to different gender, node 45 and 64 could not be grouped together. Instances in node 45 and 64 are all without heart disease, from which, we can conclude that sex is not significant in determining the presence of heart disease. However, when people do not show any symptoms of chest pain, normal (no defect), zero fasting blood sugar, but have left hypertrophy, they are very likely to not to have heart disease. When we compare nodes 45 and 1, they are all male, but when we compare weight values of the exercise induced angina attribute, instances under node 1 are more likely to have exercise induced angina other than node 45. A majority of people in node 45 have a greater risk of having heart disease, therefore, exercise induced angina is significant in determining the presence of heart disease. This conclusion is further indicated by comparing Node 63 and 1, where instances are all presented with heart disease in Node 64 when they have exercise induced angina, even with zero fasting blood sugar. Another conclusion that can be derived from Node 64 is that fasting blood sugar is not a deterministic feature in determining the presence of heart disease.

2) Two level hierarchical tree

There are six trees with two hierarchical levels. One of the trees is illustrated in Figure 17 where instances have atypical angina, which differs from all level 1 concepts presented in the previous section; therefore, it is reasonable for this tree to not to be grouped with other three hierarchical levels trees. As we can see from the diagram, when people have atypical angina, have zero fasting blood sugar, do not have exercise induced angina and do not have any defects, they are diagnosed with not having heart disease.

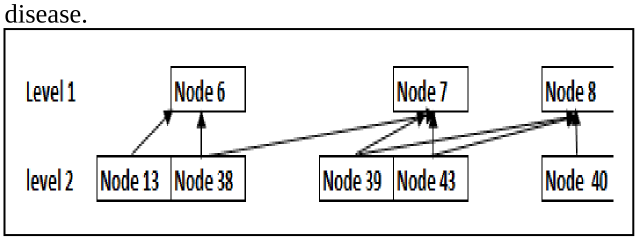

Figure 15. Trees with shared child nodes

3) Standalone nodes at level 3

There are 8 standalone nodes at level 3. For example, Node 4 at level 3, whose concept is "graphic normal", "non-angina pain", "zero fasting sugar", and "reversible defect". 7 out of 8 instances have the value 1 of the attribute "diagnosis of heart disease". When comparing this node with concepts from three level trees, Tree 1's concept (non angina pain, normal graphic and no defect) is quite similar to Node 4. All instances in Tree 1 do not have any defect, which is different from the concept reversible defect in Node 4. That is the reason why Node 4 is not grouped with nodes from Tree 1.

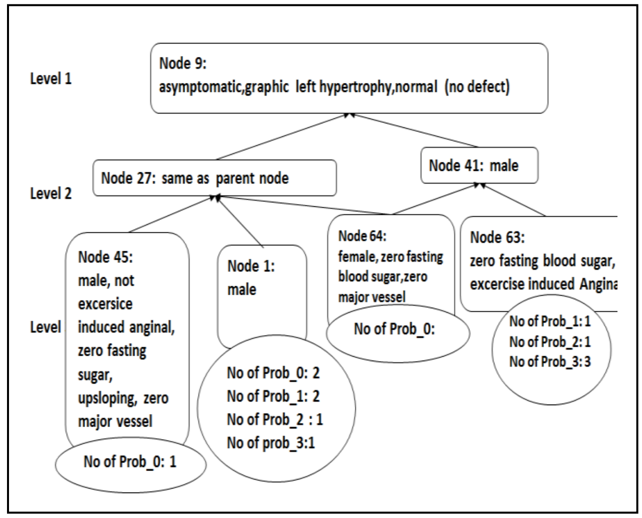

Figure 16. Three Level Concept Tree For Node 9

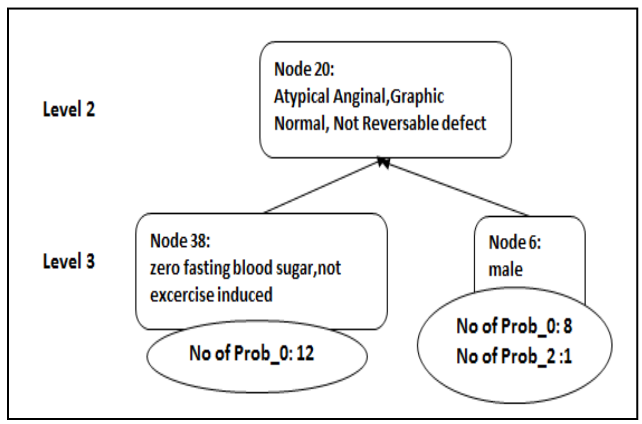

Figure 17. Two level concept tree example

V. CONCLUSION

A new model of information capture, accumulation and adaptation is described in this paper. The model is inspired by the columnar architecture of the neocortex and built using their key concepts and components, Growing SOMs, hierarchical tree structures and incremental learning. The paper describes initial results using two benchmarks data sets from the UCI repository. Although these are not time based data, the input data was divided into subsets and presented in a manner to simulate temporal inputs. The results demonstrate that the model is capable of capturing and representing multi-level concepts from the data and also has the ability to represent sub concepts with multiple parents. This provides the ability of representing a particular situation with multiple 'view points'. The purpose of the presented experiments was not to ascertain the accuracy of classification of the data by the new method. The GSOM has been utilized with many data sets in the past and has shown to be a useful data clustering and hierarchical cluster generation technique. In the described experiments we use intuitive analysis of the concepts formed by the proposed technique but also have validated these outcomes using past applications of these data sets. But the main focus was the concept formation and incremental update within an architecture based on the columnar formation of the human brain. Such an architecture was essential for the next stage of our research. The described architecture is now being used as the base for implementing cross columnar links and prediction generation. In the current proposed model (which is a key component of the data accumulation and integration model being planned), all the data attributes are processed by the GSOM in GSOM layer, while in the larger complete model, each GSOM component will process a group of relevant attributes, which is the subset of the whole attribute set of the input data. Each GSOM component at GSOM layer will be located in one column. Cross columnar links will be generated to link all columns to demonstrate the inner relationships between columns, which is the foundation for the implementation of the prediction functionality in future complete model. The work is ongoing and the base model described in the paper has provided a good foundation for a dynamic cognitive architecture which could capture sequences in data and also cross columnar relationships in data.

REFERENCES

[1] D. Alahakoon, S. Halgamuge, and B. Srinivasan, “Dynamic self-organizing maps with controlled growth for knowledge discovery”, Neural Networks, IEEE Transactions on, vol. 11, 2000, pp. 601-614

[2] D. H. Fisher, "Knowledge Acquisition Via Incremental Conceptual Clustering", Machine Learning, vol. 2, 1987, pp. 139-172

[3] E. A. Feigenbaum and H. A. Simon, "EPAM-like models of recognition and learning", Cognitive Science, vol. 8, 1984, pp. 305-336

[4] D. W. Aha, UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. Available from: http://archive.ics.uci.edu/ml 2014.03.20

[5] J. Hawkins, On Intelligence, 2^{nd} ed, Ameiran: Times Books, 2005, pp. 20-272

[6] J. Weng, “Symbolic Models and Emergent Models: A Review”, IEEE Tracnscations on autonomous Mental Development,4, pp. 29-54

[7] K. Bache and M. Lichman, UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science, 2013. Available from: http://archive.ics.uci.edu/ml 2014.03.20

[8] M. Gennari, G. Alacqua, F. Ferri, and M. Serio, " Characterization by conventional methods and genetic transformation of Neisserianceae isolated from fresh and spoiled sardines," Food Microbiol, vol. 6, pp. 61-75

[9] M. Lebowitz, "Experiments with Incre- mental Concept Formation: UNIMEM", In Machine Learning, vol. 2, no. 2,1987, pp. 103-138.

[10] J. R. Norris, "Markov Chains", Cambridge University Press, 1999.

[11] R.S.Michalski, " A Theory and methodology of inductive learning," Artificial Intelligence , vol. 20, no. 2, 1983, pp. 111-161.

[12] S. K. Chalup, "Incremental Learning in Biological and Machine Learning Systems", International Journal of Neural Systems, vol. 12, no. 6, 2002, pp. 447-465.

[13] V. B. Mountcastle, " An Organizing Principle for Cerebral Function: The Unit Model and the Distributed System," The Mindful Brain, MIT Press, 1978