The Evaluation and Comparative Study with a New Clustered Based Machine Learning Algorithm

Alauddin Alomary[1] and Mohammad Jamil[2]
[1]Department of Computer Engineering, University of Bahrain, Bahrain
[2]Department of Math and Computer, Qatar University, Qatar

Abstract: *In this paper, a clustering based machine learning algorithm called Clustering Algorithm System (CAS) is introduced. The CAS algorithm is tested to evaluate its performance and find fruitful results. We have been presented some heuristics to facilitate machine-learning authors to boost up their research works. The InfoBase of the Ministry of Civil Services is used to analyze the CAS algorithm. The CAS algorithm was compared with other machine learning algorithms like UNIMEM, COBWEB, and CLASSIT and was found to have some strong points over them. The proposed algorithm combined advantages of two different approaches to machine learning. The first approach is learning from examples, CAS supports single and multiple inheritance and exceptions. CAS also avoids probability assumptions which are well understood in concept formation. The second approach is learning by observation. CAS applies a set of operators that have proven to be effective in conceptual clustering. We have shown how CAS builds and searches through a clusters hierarchy to incorporate or characterize an object.*

Keywords: *Clustering algorithm, unsupervised learning, evidential reasoning, incremental learning, multiple inheritances, overlapping concept.*

Received December 26, 2004; accepted May 24, 2005

1. Introduction

Machine Learning (ML) can be defined as the process which causes systems to improve with experience [22]. A computer program is said to "learn" from experience *E* with respect to some class of tasks *T* and performance *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E* [14]. Interest in ML [7] increases due to the exponential growth of the amount of data and information due to the fast proliferation of the Internet, digital database systems and information systems. To automate the process of analyzing such huge data, ML becomes a crucial task. ML can provide techniques for analyzing, processing, granulation and extraction of the data [7, 10]. Also in some area, ML can be used to generate "expert" rules for the available data, especially in medical and industrial domains, where there may be no experts available to analyze data [2, 7].

ML can be either *supervised* or *unsupervised* [25]. In supervised learning, there is a specified set of classes and each example of the experience is labeled with the appropriate class. The goal is to generalize from the examples so as to identify to which class a new example should belong. This task is also called *classification*. In unsupervised learning, the goal is often to decide which examples should be grouped together, i. e., the learner has to figure out the classes on its own. This is usually called *clustering*.

In this paper, we will be concerned with unsupervised learning. We have suggested a clustering based machine learning algorithm called Clustering Algorithm System (CAS). The CAS algorithm is tested to evaluate its performance and find fruitful results. We have presented some heuristics to facilitate machine-learning authors to boost up their research works. We have taken the InfoBase of the Ministry of Civil Services to analyze our CAS algorithm with other machine learning algorithms like UNIMEM, COBWEB and CLASSIT. The proposed algorithm combined advantages of two different approaches to machine learning. The first approach is learning from examples, CAS supports single and multiple inheritance and exceptions. CAS also avoids probability assumptions which are well understood in concept formation. The second approach is learning by observation. CAS applies a set of operators that have proven to be effective in conceptual clustering. We have shown how CAS builds and searches through a clusters hierarchy to incorporate or characterize an object.

In this paper, section one presents a summary of relevant works. In section two, the proposed CAS algorithm will be described in detail. The ephemeral narration of the algorithm with its strong points will be introduced in section three. A comparison between the proposed algorithm with other relevant algorithms will be shown in section four. Finally, a conclusion will be drawn in section five.

2. Relevant Work

Many machine learning algorithms can be found in the literature [12, 20, 22, 23, 26]. These algorithms are implemented using different approaches. They may be based on heuristic search [26], inductive logic programming, Bayesian approach [6], neural networks, and conceptual clustering [4, 22]. In this paper we are concerned with conceptual clustering algorithms. Some well known clustering based algorithms found in the literature includes UNIMUM, COBWEB, CLASSIT, CLASSWEB, CLUSTER/2 and WITT.

UNIMEM algorithm [20] is designed for experiments on acquisition and use of concepts for tasks such as natural language understanding. It organizes the knowledge from instances observed into a concept hierarchy. However, UNIMEM has some problems such as top nodes are updated regardless of whether they match the instance observed which leads to a bias toward concepts that are represented with a larger number of instances. Also, despite the fact that UNIMEM implements a form of forgetting, UNIMEM stores training instances and thus the hierarchy can become very large.

COBWEB algorithm [12] is designed based on work done in cognitive psychology. It also uses a predictive score and introduces three additional indicators to sort the instances observed through its concept hierarchy. In COBWEB, the processes of learning and classification are done in the same time and as the instance is sorted along the hierarchy nodes, the nodes themselves are updated. COBWEB also has better defined procedures to apply the learning operators. The nodes are updated based on category score. The problems with COBWEB are that COBWEB stores all instances observed and has tendency to over fit data.

CLASSIT algorithm [12] is an extension of COBWEB that handles both symbolic and numeric attributes. CLASSIT uses artificial domains that involved four separate classes, each differing in their values on four relevant numeric attributes. However, the domains varied in the number of irrelevant attributes-which have the same probability distribution independent of class-from zero to sixteen. All domains had small but definite amounts of attribute noise, and training instances were unclassified. The performance task involved predicting the numeric values of single relevant attributes omitted from test instances, and the dependent measure was the absolute error between the actual and predicted values. In CLASSIT, irrelevant knowledge could slow the learning rate of analytic learning approaches by producing misleading explanations or making derivations intractable. Techniques for selecting among competing explanations and selecting likely search paths could play a similar role to the evaluation function that CLASSIT uses to ignore irrelevant attributes.

CLASSWEB [20] is the combination of COBWEB (building concept hierarchies, symbolic) and CLASSIT (building concept hierarchies, numeric) and back propagation (sub-symbolic).

In CLUSTER/2 [22] and WITT [26] the cost of incorporating a single object is significantly more than rebuilding a clustering tree for each new object using search-intensive methods that have a polynomial or exponential cost.

3. Proposed CAS Model

We have proposed a clustering based algorithm called CAS. In this section, the detailed description of CAS algorithm will be introduced.

3.1. Employment and Visa Problems

In the state of Qatar, the Ministry of Civil Services is responsible for assigning jobs for Qatari and Non-Qatari. The InfoBase of the Ministry of Civil Services is selected and used to test and compare our model and find interesting reports about appointment in government and private job for Qatari and non Qatari. Accordingly, visa is issued for a particular nationality for non Qatari. The data inserted in series of instances taken from samples of Ministry of Civil Services InfoBase. Instance has a person with feature supplied by nationality, for example, [Qatari, non Qatari] CAS consistently converges on the tree of Figure 1.

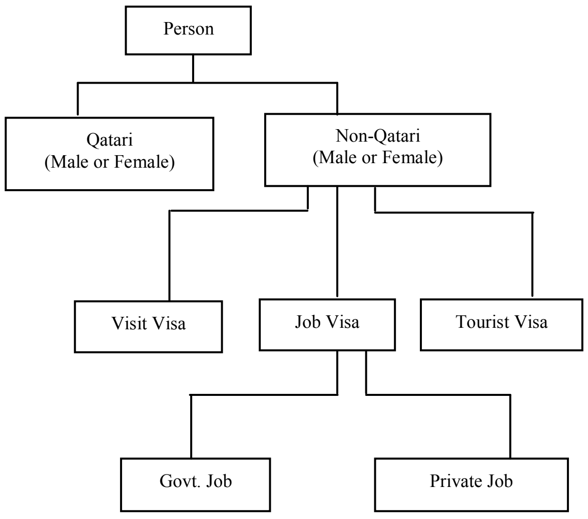

Figure 1. CAS top level clusters.

CAS organizes instances into a hierarchy of concepts based on likeness [17]. In sample data, the countries are classified or clustered with respect to the similarities based on their features. The algorithms by observation, incremental in growth of the concept hierarchy, handle a large set of input instances and are able to make inference about queries made to the system based on partial matching. When a new instance is supplied to algorithm then, it will try to match the instance to the existing concept hierarchy otherwise, it creates a singleton class of its own.

3.2. CAS Operators

CAS forms a clustering tree and its node (cluster) [17] contains frequency information that epitomizes objects within that cluster. CAS uses hill-climbing search to place in a most appropriate node in the tree for a given object. The hierarchal clusters space uses theses operators.

1. *Create* new cluster.
2. *Update* an existing cluster with an object.
3. *Fuse* two clusters into one.
4. *Divide* cluster.

The create operators automatically allow to amend recently created cluster in the existing clustering depending on which clustering is preeminent with respect to the best estimated rules. If a new cluster is created, then system identifies this singleton cluster as the object, otherwise object placed in the existing cluster in the opposite place.

If a new object is added then update operators allow updating the existing cluster with the best estimate rule that consists of set of heuristics that update frequency distributions. If initial input objects are non-representative of the entire population, then create and update operators can forms a hierarchies with poor predictive ability. To avoid this we use two other operators fuse and divide that allow the bi-directional movement within hierarchy.

The fuse operator combines two clusters into new one after combining their characteristics values frequencies of clusters being fused. The divide operator deletes a cluster on a level of *n* clusters and promotes the children of the deleted cluster so that the level now has $n + m - 1$ clusters, where *m* is the number of children of the deleted cluster. If the situation does not suit with the existing cluster then the divide operator may undo the effect of the fuse operator and vice versa. To place newly created cluster in appropriate place, CAS performs search in the cluster hierarchy using above four mentioned operators.

3.3. Formation of Hierarchy

In this subsection, we will describe how the CAS forms the hierarchy and performs search in the cluster hierarchy. The process works by applying the following steps at each consecutive level of the hierarchy.

Step 1: An object, *O*, is presented to be clustered into the clusters hierarchy. The clustering hierarchy we have is either:

```
1.1. If consists of at least one level (i., e., the
     hierarchical structure has a root with at least two
     children).
     If the root can incorporate O, then
       update frequencies at the root and go to step 2,
       taking the root to be the current cluster,
     Else
       Go to 1.2 of step 1.
1.2. If consists of only one node, T, then the best
     estimate rule is applied to decide.
     If whether T may incorporate O then
       call update operator and terminate,
     Else
       Create a new node G.
       Where G is a generalization of T and O. The
       nodes T and O are inserted as children of G.
       At the end of this process we have a tree with
       G as its root and the process terminates.
1.3. If empty (i. e., O is the first object in the object
     set), in which case CAS creates a terminal node, T,
     corresponding to O, and the routine terminates.
```

Step 2: Among the children of the current cluster, identify the one (if any) that has the highest likelihood which is computed according to the best estimate rule. According to the best estimate rule, no child is identified if each child differs from *O* in at least one characteristic value. However, a human expert can override this result and base the estimates of likelihood only on the number of characteristic values that are common to *O* and the child. *O* is incorporated into the clustering based on one of the following:

```
2.1. If no child of the current cluster has been
       identified then
         make O a child of the current cluster
         by applying the cluster creation operator,
     Else
     If current cluster is a terminal node (i. e., a
       micro-cluster) then terminate occurs,
     Else
       Creation operator is applied, and in this case
       the system considers creation of possible new
       clusters created by generalization of each child
       and O, and selects the best of such candidate
       clusters by using the best estimate rule. From
       within the create operator, CAS now applies
       the fuse operator to each of these selected
       clusters and O, and terminates.
2.2. If a child of the cluster is identified as a best host
       cluster for O then
         incorporate the object into the child by
         applying the update operator to update the
         values of the child that are present in O.
         After this update operation, CAS considers
         the possible deletion of the current cluster
         by applying the divide operator to the
         current cluster and its child. Whether or not
         division takes place, CAS treats the child as
```

the current cluster in a recursive call of Step 2. By using this procedure, the system tends to converge on clustering hierarchy in which the high levels contain well separated clusters as a result of entropy maximization by the use of the best estimate rule. Further down towards the leaves, the clusters tend to overlap and to be more diffuse.

3.4. CAS Data Structure

Representation of knowledge has been done in the form of the hierarchy and child parent concept. We have used the tree type data structure.
We will show how CAS incorporates these to form its data structure.

The CAS nodes are of three types:

- Cluster nodes (C), consists of set of all objects.
- Characteristic nodes (A), consists of characteristic applicable to a cluster and list of value nodes.
- Value nodes (V), consists of an integer values and frequency.

Nodes consist of two types of links.

- Inter-cluster links:

  a. Aggregation Link (RL) is top to bottom link.
  b. Characterization Link (IL) is bottom to top link.

- Intra-cluster links:

  a. Characteristic Link (AL) cluster links with characteristic node.
  b. Value Links (VL), characteristic node links with its value node.

Cluster nodes generally consist of the following information:

- Number of objects, a numeric integer value.
- Object sets, a set of theoretic representation of objects.
- Characterization Links (IL).
- Aggregation Links (RL).
- Characteristics nodes (A).
- Value node (V).
- Characteristic Links (AL).
- Value Links (VL).

A characteristic node contains the following information shown in Figure 2.

Characteristic Ai
Value nodes list
Frequency
Address of next characteristic

Figure 2. Characteristic node.

And value node contains the following information shown in Figure 3.

Value Vi
Frequency
Address of next value

Figure 3. Value node.

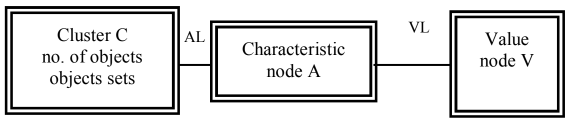

Figure 4. Cluster node

Since characterization is the dual of aggregation therefore RL links are top down links while IL links are bottom-up links. If cluster E is a parent of cluster F in the clusters structure, then there is a bottom up link IL from cluster F to cluster E and a top down link RL from cluster E to cluster F.

If characteristic node A_m is attached to cluster C_m, then A_i must have at least one value node attached, otherwise, node A_m is detached and A_i becomes a non-local characteristic of cluster C_m. After the characteristic A_m has been detached, A_m becomes a default characteristic, which is still applicable to cluster C_m through the inheritance mechanism. In this situation, the inheritance mechanism preserves characteristic and its value.

3.5. Characterization

Characterization is the conceptual description of a cluster on its characteristics. We say that a characteristic of a cluster is local to that cluster if it is not inherited from an ancestral cluster. On the other hand, a characteristic that is inherited from an ancestral cluster is said to be non-local to the cluster.

In our work on conceptual clustering, the task of characterization involves not only determination of local characteristics of the cluster, but also involves a determination of non-local characteristics that apply to the cluster through a referencing mechanism.

We introduce referencing (inheritance) in characterization whereby the clustering system infers characteristics of a cluster from characteristics of its ancestors. In practice a cluster must have a *reference* to where non-local characteristics are to be found. For instance, if the system knows that "*every non-Qatari has proper visa"* then given "*Egyptian is a non-Qatari*" it may infer that "*Egyptian has visa*". Reasoning such as this is called default reasoning. If local characteristics are not available, then our system searches for characteristics attached to clusters that lie above in the clusters structure.

Let C be any cluster, which may be ontological or simple. If value V of characteristic A is a local characteristic value of one of C's ancestor clusters but not of C itself, then we say that cluster C *inherits* the value V of characteristic A. On the other hand, if a cluster C *inherits* the value V of a non-local characteristic A, where A is local to more than one of

C's ancestors and there are references to all C's ancestors, then we say that we have a multiple referencing case in which cluster C has *multiple references* to value V of characteristic A.

The advantage of using the referencing mechanism is that in effect it provides a virtual copy of the description of a cluster so that there is no need to make an actual copy as in [5, 9]. Having a virtual copy reduces the memory requirement quite considerably compared with clustering systems, which always make an actual copy of the description for each cluster. Furthermore, if the multiple referencing mechanism places an object into more than one cluster, then these clusters overlap (i. e., they do not form disjoint partitions over the objects). In cluster analysis [1, 8], this phenomenon is called clumping. An advantage of having multiple referencing is that overlapping clusters may describe the data more accurately than disjoint clustering. Moreover, clumping introduces flexibility into the search for useful clusters [19, 20].

An unusual feature of the present work is that the memory requirement is considered as part of the conceptual clustering problem. This was motivated by finding experimentally that existing cluster analysis systems soon run out of memory, after processing only a few hundred objects.

3.6. Aggregation

Aggregation is the problem of distinguishing subsets of an initial object set. In other words, aggregation is the formation of set of classes, each defined as an extensionally enumerated set of objects. For the aggregation problem, an object is a description consisting of a set of characteristic value pairs, and the task is to find a cluster that best matches this description. Aggregation can be regarded as a general form of pattern matching in which the set of patterns to which an input pattern is to be matched are organized in a hierarchy. Matching an input pattern A with a target pattern A_j involves matching characteristics that appear in A with characteristics local to A_j as well as to characteristics that A_j refers to amongst its ancestors.

3.7. Time and Space Complexity

The strategy by which CAS finds a solution is by viewing characterization and aggregation as two separate but interconnected processes. In each process, CAS searches for a solution in a single direction, which is top to bottom in aggregation, or bottom to top in characterization. The direction of search is determined by the partial ordering. If during aggregation the system is unable to find a characteristic value, then it has to suspend the aggregation process whilst it tries to estimate the unknown characteristic value by activating a characterization process. This involves a search that is guaranteed to terminate by Well-Formedness Rule (WFR) [17]. Once the value of a characteristic has been found, then aggregation is reactivated, and proceeds away from the root cluster until the object has been dealt with as explained previously.

Because characterization search is bound to terminate, and because the aggregation process is a one-way trip through the hierarchy in a direction away from the root, there is no possibility that the entire process will become trapped in an infinite loop of fruitless repetition. Indeed the total time for incorporating a new object into the clusters hierarchy is proportional to the depth of this hierarchy.

If we assume that c is the average branching factor of the tree and that N is the number of objects already classified, then an approximation for the average depth of a leaf is *($\log_C N$)*. Furthermore, let A be the number of defining characteristics and V be the average number of values per characteristics.

In the clustering process, comparing an object and a current cluster, appropriate frequencies are incremented and the entire set of children of the current cluster are evaluated by best estimate rule. The cost depends linearly on A, V and c, so we can say that the process has complexity O ($c\ A\ V$). This process has to be repeated for each of the c children. Hence, comparing an object to a set of siblings requires O ($c^2\ A\ V$) time, in general, clustering proceeds to a leaf, the approximate depth of which is ($\log_c N$). Therefore, the total number of companions necessary to incorporate an object is approximately O ($c^2 \log_c N\ A\ V$).

The branching factor is not bounded by a constant as in CLUSTER/2 algorithm, but it is dependent on regularity of the environment. In practice the branching factor of trees generated by CAS varies between two to six. This range agrees with the intuition [22] that most good clustering trees have small branching factors, and lends support to bounding the branching factor in their system. By any means the cost of incorporating a single object in CAS is significantly less than rebuilding a clustering tree for each new object using search-intensive methods that have a polynomial or exponential cost, as in WITT [15] or CLUSTER/2 [21].

We represent clusters C in n-dimensions space where n is the number of characteristics. Each dimension of the space corresponds to an applicable characteristic and the marginal correspond to F ($C\ [A, V]$), which is the number of objects in the sub-cluster C_v of the clusters set C having the value V for characteristic A. The extent of a dimension is given by the number of distinct values of the characteristic. The points in the space denote the number of objects in the cluster that have the appropriate combination of characteristics values. In the two dimensional space, if two characteristics A_1 and A_2 are applicable to some cluster C. Furthermore, the system knows all the F (C [A_1, V_i])s and F (C [A_2,V_j])s where the V_i and V_j are the values of the characteristics A_1 and A_2 respectively. Looking into this procedure, the values of F (C [A_1, V_i][A_2, V_j])s will be estimated by the system itself.

We can calculate e_{ij} like below:

$$R_i = F(C[A_1, V_i])$$
$$C_j = F(C[A_2, V_j])$$
$$N = \sum_{i=1}^{n} R_i = \sum_{j=1}^{m} C_j = F(C)$$
$$e_{ij} = F(C[A_1, V_i][A_2, V_j])$$

e_{ij} is cluster and it is a Cartesian product of the above space and R_i row sum and C_j is the column sum. The value of R_i and C_j is know to the system but still e_{ij} are unknown and need to be estimated. In other words, the system needs to determine the most probable count configuration indicated by the following information:

$$\forall i = 1..n \sum_{j=1}^{m} e_{ij} = R_i$$
$$\forall j = 1..m \sum_{i=1}^{n} e_{ij} = C_i$$
$$\sum_{i=1}^{n} \sum_{j=1}^{m} e_{ij} = N$$

We can recast our problem as follows: Consider a distribution of *N* district objects onto a two dimensional space of clusters in a manner consistent with the constraints imposed by available information. We interpret e_{ij} as the specification of the number of objects placed in the ijth cluster. In terms of this formal definition, the number and identity configurations are interpreted as follows: The count configuration specifies the number of objects placed in each cluster (i. e., points) in the clusters space (i. e., Cartesian product space); the identity configuration is the complete result of such distribution including the identity of the objects in each cluster.

The feasible identity configurations are only those, which satisfy the constraints imposed by row and column sums. As explained previously, these configurations are equally likely with respect to the system's knowledge. Following the principle of insufficient reason, the only rational assumption is that all feasible identity configurations are equally probable. We have seen the most probable count configuration will be supported by the greatest number of feasible identity configurations.

This problem is an example of constraints maximization problem and can be solved using the technique of Lagrange multipliers. We have seen that solution given by

$$\forall(i = 1..n, j = 1..m) e_{ij} = \frac{R_i x C_j}{N}$$

satisfies the condition of maximality.

That is, if we consider all possible ways of distributing n district objects into a 2-dimensional space of clusters, subject to the constraint imposed by row and column sums R_i's and C_j's, then the distribution of objects wherein each point e_{ij} contains R_i x C_j / *N* objects will occur more often than any other distribution. The clusters space for Qatari and expatriate population cluster with respect to employment and gender of person is shown in Figure 5. Here one unit equal to 6 thousands persons.

Qatari

	Male	Female	Row Sum N = 100
Gove. Job	?	?	60
Priva Job	?	?	40
Col Sum N = 100	80	20	100

Non Qatari

	Male	Female	Row Sum N = 600
Gove. Job	?	?	240
Priva Job	?	?	320
Col Sum N = 600	480	120	600

Figure 5. Matrix representation of clusters.

The solution implies that if we consider all possible ways of assigning employment and gender to 100 Qatari units and 600 non-Qatari units while honoring the constraints that 48 Qatari are govt. job while 32 are private job, 80 Qatari are male while 20 are female, 240 non-Qatari are govt. job while 360 are private job, and 480 are male while 120 are female, then the distribution of Qatari and non Qatari in Figure 6 will occur more often than any other distribution.

Qatari

	Male	Female	Row Sum N = 100
Gove. Job	48	12	60
Priva Job	32	8	40
Col Sum N = 100	80	20	100

NonQ atari

	Male	Female	Row Sum N = 600
Gove. Job	192	48	240
Priva Job	288	72	360
Col Sum N = 600	480	120	600

Figure 6. Distribution of objects.

Thus, a rational system would decide that on the basis of the available information, the most probable distribution of Qatari and non Qatari is as given in Figure 6. Consequently, the system will identify a female and private job sample to be a member of the cluster non Qatari as most probably there are 72 non Qatari that meet this description as against only 8 Qatari. Where a male and govt. job will be member of cluster non Qatari 192 units meets with 48 Qatari millions. So, visa will be issued to 192 units male expatriates for the govt. jobs and 288 units private jobs or businessmen.

The best estimate rule can be extended to higher dimensions and general form of the solution will be

$$e_{ij} = \frac{(R_i x C_j)}{N}$$

In terms of the representation language, this rule will be based on the knowledge of number of objects having value V_n for characteristic A_n, then the best estimate of the number of objects having value V_n for characteristic A_n is given by

$$\mathrm{N\ x} \prod_{i=1}^{n} \frac{F(Ai, Vi)}{N}$$

The approach we used to compute the best estimate rule is the maximum entropy approach that is equivalent to the basic probabilistic approach. Under the maximum entropy approach, each piece of information is considered as a constraint. These constraints are used to determine the most probable count configuration of the domain, and all unknown probabilities are computed with reference to this count configuration.

4. Ephemeral Narration of Algorithm

The Clustering Algorithm System (CAS) is rooted on evidential reasoning utilizing the assumption of reasonableness. Evidential clustering property matches an instance to a concept after looking almost conceivable concept from the set of discretion. The amount of reasonableness of mapping an instance to a concept is estimated with respect to the knowledge stored in the concept hierarchy. The evidential information is represented in the form of relative frequencies in the representation language of the system. Clustering hierarchy is built incrementally, with each cluster node containing frequency information that maps an instance to that cluster. The representation language takes into account the current ignorance while incorporating an instance into the cluster. Since it is based on evidential reasoning it is theoretically stronger than COBWEB and CLASSIT in terms of its representation of ignorance [17]. It combines a number of different paradigms such as constraint satisfaction, evidential reasoning, and inference maximization and entropy maximization. Combination of evidence is based on best estimate rule using the notion of maximum entropy [24].

CAS resolves the problems of multiple inheritance and exceptions. Exceptions is a property wherein a feature may hold true for most instances of a concept, but may not hold for instances of the concept's sub-generalization. Using the inheritance property of the cluster hierarchy, we infer features of a concept based on the features of its ancestors. In some domains, the clustering problem would result more naturally into multiple hierarchies in which a concept may have more than one parent, each belonging to different hierarchies. The concept inherits the features of both parents. This may lead to conflicting information if the features of the parents of a concept node contradict.

It is similar to COBWEB and CLASSIT in that it uses a similar hill-climbing search through the hierarchy in mapping an instance to a cluster. It uses the same four clustering operators with the only difference of cluster evaluation function. CAS update operation is based on a set of heuristics that update the frequency distribution. Upon reaching the most likely incorporating cluster, CAS uses the best clustering estimate rule to select one of the four operators. Maximum entropy is used to compute the best estimate rule [24]. This is equivalent to the basic probabilistic approach. With maximum entropy, each instance is considered as a constraint. These constraints are used to determine the most probable clustering configuration of the domain. All unknown probabilities are computed with reference to this configuration. Maximum Entropy formulates a precise way of estimating unknown probabilities. Unlike COBWEB and CLASSIT, which update probabilities and standard deviation respectively, CAS updates the frequencies.

5. Comparative Results

5.1. Accomplishment Consequence

The four algorithms, UNIMEM, COBWEB, CLASSIT and CAS, were implemented and tested using the Civil Service Department InfoBase mentioned in section (2.1). These algorithms build a concept hierarchy with their knowledge representation based on the inputs that arrive. Each concept is a node with a combination of 2 lists, one list maintains the list of visa of instances, which are added to the concept, and the other list maintains the list of features. An instance contains a set of features, and each feature is a representation of attribute and its value.

instance = [set of features]
feature[attribute, value]

At each node UNIMEM records feature confidence values. In the COBWEB, cluster is modified upon conditional probability. The conditional probabilities of each feature attribute stored in a concept is calculated based on the count of number instances stored under the node possessing that attribute value, and the total number of instances stored under the node. CLASSIT uses mean and standard deviations of the feature and real values. Our suggested algorithm is based on the conceptual clustering [4, 11, 17]. CAS models associates relative frequencies with evidential information and solves the problems of multiple inheritance and exceptions. It avoids probability assumptions that do not adequately reflect ignorance.

5.2. Assessment Criteria

The algorithms, which we have evaluated, differ in terms of mechanism of search, storage and retrieval of knowledge from hierarchy with the complexity of algorithm. We have used certain criteria to evaluate the three different machine learning algorithms with our CAS algorithms [13, 20, 19, 16]:

- Overlapping concepts: A concept can have more than one instance.
- Multiple inheritances.
- Knowledge representation in the concept hierarchy.
- Including bi-directional operators that reverse the effects of learning if the new instance suggests the need, i. e., including operators in the machine learning algorithms for not only creating new concepts but also for deleting a concept if the new instances suggests the need. This is equivalent to hill climbing with the effect of backtracking. This leads to better performance and efficiency.
- Classification scheme: Which branch will be allocated for a new instance? Which place will be allocated either leaf or middle of hierarchy? Then what will be the performance of the hierarchy?

5.3. Comparative Summary

The algorithms learn incrementally through observation of positive instances and also handle as well as update large input as a new instance arrives and are capable of learning multiple concepts. They generate a hierarchy of instances with set of attribute value pairs by different methods using conceptual clustering. Like rule of class and sub class in the hierarchy, higher node represents more general concepts whereas lower nodes represent sub-generalizations of the higher-level nodes. It means that the children have more specific concepts than parent node. In CLASSIT system we find that concepts lower in the hierarchy have attributes with lower standard deviations.

In our suggested algorithms CAS and in COBWEB the new instance input to the system are stored only at the terminal nodes of the concept hierarchy. This works healthy in noiseless, token domains, but it tends to overfill the data in noisy or numeric domains for which concept pruning is required. The performance decreases in noisy domain because the system puts together thoroughgoing decision trees. To repossess from non-optimal hierarchy structure, the algorithm provides two additional operators, node merging and node splitting.

As a new instance arrives, it stores in hierarchy but in CLASSIT and UNIMEM it need not be a terminal node as CAS and COBWEB do. UNIMEM is able to prune or unlearn a concept if the new instances suggest the need to do so; i. e., it is able to delete an overly specific concept. CLASSIT does not retain every instance input to the system. Such pruning, by forgetting certain instances, leads to better performance and efficiency.

All algorithms except UNIMEM use a function or estimate rule to place a new instance in hierarchy whereas UNIMEM uses depth first strategy. Search in all algorithms is better as compared with UNIMEM but to place a new instance is time consuming. Also UNIMEM behaves in an unpredicted manner as hierarchy grows or shrinks.

A comparison among the suggested algorithm and other relevant algorithms is conducted and summary of the main strong points of our algorithm as compared with other algorithm is shown in Table 1.

Table 1. Summary of the main strong points of CAS algorithm as compared with others.

S. No	Characteristics	UNIMEM	COBWEB	CLASSIT	CAS
1	Concept Description	Terminal and non-terminal node	Terminal node	Terminal and non-terminal node	Terminal node
2	Attribute Value	Numeric nominal single and multi values sets	Any nominal single valued	Real value	Nominal single values
3	Instances Handling with Missing Values	Yes	Yes	Yes	Yes
4	Overlapping Concept	Yes	No	No	Yes
5	Hill Climbing Search	Yes	Yes	Yes	Yes
6	Concept Deletion	Delete an overlay specific concepts	No	No	No
7	Concept of Maximum Entropy	No	No	No	Yes
8	Sensitivity to the Order of Input	Yes	Merging & Splitting operators used recovers sensitivity		Fuse & Other operators used
9	Multiple Inheritance	No	No	No	Yes
10	Exception Handling	Yes	No	No	Yes
11	Prediction Ability	No	Yes	Yes	Yes
12	Ignorance Representation	No	No	No	Yes
13	Overlapping Domain	Yes	No	Yes	No

6. Conclusion

The CAS model associates relative frequencies with evidential information and solves the problems of multiple inheritance and exceptions. It avoids probability assumptions that do not adequately reflect ignorance. CAS has the following advantages over other machine learning algorithms:

1. It is stronger than COBWEB and CLASSIT machine learning algorithms in terms of its representation of ignorance.
2. It combines a number of different paradigms such as constraint satisfaction, evidential reasoning, and inference maximization and entropy maximization. Combination of evidence is based on best estimate rule using the notion of maximum entropy.

References

[1] Anderberg M. R., *Cluster Analysis for Applications*, Academic Press, New York, 1973.

[2] Aseltine J. H., "An Incremental Algorithm for Information Extraction," *in Proceedings of the AAAI'99 Workshop on Machine Learning for Information Extraction*, 1999.

[3] Bay S. D., "Combining Nearest Neighbor Classifiers Through Multiple Feature Subsets," *in Proceedings of the International Conference on Machine Learning*, Morgan Kaufmann Publishers, Madison, Wiscanson, 1998.

[4] Biswas G., Weinberg J., Yang Q., and Koller G., "Conceptual Clustering and Exploratory Data Analysis," *in Proceeding of the 8th International Workshop on Machine Learning*, Morgan Kaufmann, Los Altos, California, pp. 591-595, 1991.

[5] Brachman R. J., and Schmolze J. G., "An overview of the KL-ONE Knowledge Representation System," *Cognitive Science*, vol. 9, no. 2, pp.171-216, 1985.

[6] Cheng J. and Druzdzel M. J., "AIS-BN: An Adaptive Importance Sampling Algorithm for Evidential Reasoning in Large Bayesian Networks," *Journal of Artificial Intelligence Research*, vol. 13, pp. 155-188, 2000.

[7] Cios K. J., and Moraes I., "ALFS: An Inductive Learning Algorithm," *Kybernetes,* vol. 20, no. 3, pp. 19-30, 1991.

[8] Everitt B., *Cluster Analysis*, Heinemann Educational Books, London, 1980.

[9] Fahlman S. E., *NETL: A System for Representing and Using Real World Knowledge*, MIT Press, London, 1979.

[10] Faure D. and Nédellec C., "Knowledge Acquisition of Predicate-Argument Structures from Technical Texts Using Machine Learning," *in Proceedings of Current Developments in Knowledge Acquisition* (EKAW'99), pp. 329-334, 1999.

[11] Fisher D., "Knowledge Acquisition Via Incremental Conceptual Clustering," *Machine Learning*, vol. 2, no. 2, pp. 139-172, 1987.

[12] Gennari J. H., Langley P., and Fisher D., "Models of Incremental Concept Formation," *Artificial Intelligence*, vol. 40, pp. 11-61, 1989.

[13] Gluck M. and Corter J., "Information, Uncertainty, and the Utility of Categories," *in Proceedings of the 7th Annual Conference of the Cognitive Science Society*, Irvine, CA, pp. 283-287, 1985.

[14] Haipeng G., "Algorithm Selection for Sorting and Probabilistic Inference: A Machine Learning Approach," *PhD Thesis*, Department of Computing and Information Sciences, College of Engineering, Kansas State University, 2003.

[15] Hanson S. and Bauer M., "Conceptual Clustering, Categorisation, and Polymorphy," *Applied Artificial Intelligence,* vol. 40, no. 1, pp. 11-61, 1989.

[16] Ioannidis Y. E., Saulys T., and Whitsitt A. J., "Conceptual Learning in Database Design," *ACM Transactions on Information Systems*, vol. 10, no. 3, pp. 265-295, July 1992.

[17] Jamil M. S., "Learning Algorithm Model: Clustering Algorithms System," *PhD Thesis*, VKS University, India, 2005.

[18] Keogh E., and Pazzani M., "Learning Augmented Bayesian Classifiers: A Comparison of Distribution-Based and Classification-Based Approaches," *in Proceedings of the 7th International Workshop on AI and Statistics*, Ft. Lauderdale, Florida, pp. 225-230, 1999.

[19] Lebowitz M., "Concept Learning in a Rich Input Domain: Generalization-based Memory," *Machine Learning*: *An Artificial Intelligence Approach*, vol. 2, pp. 193-214, 1987.

[20] Lebowitz M., "Experiments with Incremental Concepts Formation: UNIMEM," *Machine Learning*, vol. 2, no. 2, pp. 103-138, 1987.

[21] Michalski R. S. and Stepp R. E., "How to Structure Structured Objects," *in Proceedings of the International Workshop of Machine Learning*, pp. 156-59, 1983.

[22] Michalski R. S. and Stepp R. E., "Learning from Observation: Conceptual Clustering," in Michalski, R., Carbonell J., and Mitchell T. (Eds), *Machine Learning: An AI Approach*, Chapter 11, pp. 316-364, 1983.

[23] Quinlan J., *Programs for Machine Learning*, Morgan Kaufman, 1993.

[24] Shastri L., "Semantic Networks: An Evidential Formalization and Its Connectionist Realization," *Research Notes in Artificial Intelligence*, Pitman, London, 1988.

[25] Tan A. C. and Gilbert D., "Machine Learning and Its Application to Bioinformatics: An Overview," *Technical Report*, Bioinformatics Research Centre, Department of Computing, University of Glasgow, United Kingdom, 2003.

[26] Thrun S., Bala J., Bloedorn E., Bratko I., Cestnik B., Cheng J., Jong K. D., Dzeroski S., Hamann R., Kaufman K., Keller S., Kononenko I., Kreuziger J., Michalski R. S., Mitchell T., Pachowicz P., Roger B., Vafaie H., de Velde W. V., Wenzel W., Wnek J., and Zhang J., "The MONK's Problems: A Performance Comparison of Different Learning Algorithms," *Technical Report CS-CMU-91-197*, Carnegie Mellon University, December 1991.

Alauddin Alomary received his BSc in electronic and communication engineering from the University of Mosul, Iraq in 1980, his Master degree in communication engineering from University of Technology, Baghdad, Iraq in 1986, and his PhD in system and information engineering from Toyohashi University, Japan in 1994. Currently, he is an associate professor at the Department of Computer Engineering, College of Information Technology, University of Bahrain. His research interests include artificial intelligence, web intelligence, computer networks and ASIC systems design using VHDL. He has been actively involved in many research projects and published more than 25 papers. He is a member of IEEE and the Japanese Information Processing Society.

Mohammad Jamil received his MSc from Aligarh Muslim University with first class first position, then he received another Master in computer science and application with first class first position from Aligarh Muslim University, India and awarded gold medal from the president of India for securing first position top rank. He has also submitted PhD in the areas of artificial intelligence from VKS. University. Currently, he is working as a senior lecturer in the Department of Computer and Math, Foundation Department, University of Qatar. He has over 16 years of diversified experience in computer science and information technology area and published more than 16 papers and some are under process.

.