Hierarchical Multi label Classification of Set-Valued Data

Jijiya R P, Aswathy V Shaji

Abstract—Nowadays we are dealing with large amount of data consisting of set valued attribute with class labels arranged in hierarchical order and more than one class label for each single instance. Classification of this kind of data is a challenging task. We address this problem using classification algorithm (decision tree) and hierarchical multi label classification using Ant Colony Optimization. Our proposed system is a major extension of an ant colony optimization algorithm for generating rules. The paper also evaluates the performance by comparing with one of the existing classification algorithms (decision tree) in the area

Index Terms—Classification, Ant Colony Optimization, hierarchical multi labeled classification.

1 Introduction

As we are dealing with huge amount of data now a days manual categorization is impractical. This is why machine learning experts have explored the possibilities of automating the process by classifying the data based on the hidden knowledge within the data. The idea is that we will classify large set of training data and we impose the rules generated from this dataset to classify the remaining unlabelled data.

Our work explores a new domain for classifying data with set-valued attributes and hierarchical multi labeled classes. Class labels of a data set can be of single-label, multi label and hierarchical where the main goal of a single labeled data set is to predict a single class. In Multi label, two or more than two classes are to be predicted. In Hierarchical, class label values can be of hierarchical structure.By contrast, in our proposed system data set consists of attribute with more than one value and class labels are of hierarchical structure with more than one value to be predicted. Because of increasing interest in large datasets with hierarchical class label having multiple classes, this is an active research area, particularly in the areas of text mining and bioinformatics. In text categorizarion we can organize these data set into hierarchical structure. This is very common nowadays. Figure 1 shows hierarchical structure of web repository

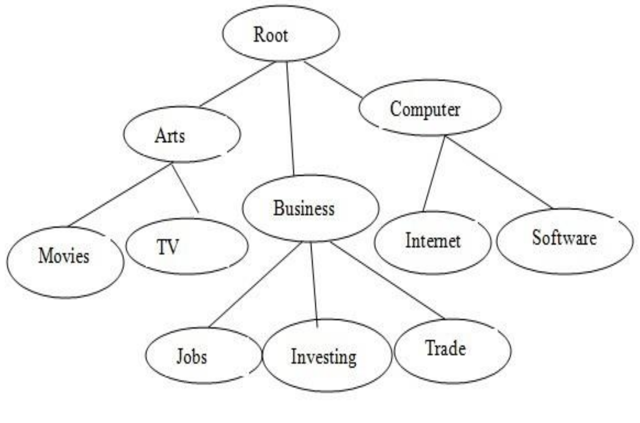


Fig. 1. Hierarchical Structure of a web repository

There are different approaches for classifying data such as decision tree, bayesianclassification, neuralnetwork, geneticalgorithm,rule referring. They are capable of helping people to make good decisions. But in these cases, there are some short comings like, unintelligible results, over-fit rules and difficulties in being applied on distributed simulation. Here we present the major findings of comparative study of two methods namely, decision tree and ant colony optimization.

Decision tree is a tree structure that classifies instances by sorting them based on attribute selection methods. Each node represents feature in an instance to be classified each branch represents a value that the node can assume. Starting at the root node and sort based on their feature values. Each terminal nodes holds class label. It represents outcome of the test. The heuristic measure that we used here is modification of traditional entropy-based method called as hentropy. In this paper we evaluated the performance of decision tree using Shannon entropy and Renyi entropy. Shannon entropy is the limiting case of Renyi entropy. Ant colony optimization(aco) is a system used for solving optimization problem. It is based on agents called ants which deposits some chemicals called pheromone in the ground which leads to make a path by the trail of this substance. Communication among individuals or between individuals and the surroundings is based on the use of these pheromones. The design of aco is based on certain aspects such as heuristic function that measures the significance of items thatcan be added to the rule set,ruleupdation which specifies how to fine tune pheromone trail, probabilistic transition function based on the value of heuristic function and pheromone trail. Our work is an extension of an ant colony optimization which generates rules of the form:

IF <condition> T HEN <predicted class> [1]

means, if an example in the dataset satisfies any of the condition in the antecedent part of an example then that example is assigned the class that is predicted by the consequent.In our case, some of the example in the dataset may contain a set of values. Also, consequent part consists of more than one value with classes of hierarchical relation. This extension requires major re-design of the algorithm which we will discuss later. Then we will make a comparative study on the performance of our algorithm with other existing algorithms.

2 Related work

Recently in modern application of machine learning the advancement of computing has been increased in terms of cost, processing huge amount of data and also the computing power. Machine learning is one of the commonly used methods. The data provided to the system during training are the primary source of knowledge which is learned by the system. Classification is one of the most important learning tasks. Most of the work in classification has been motivated by text categorization. We focus on explaining works based on decision tree and Ant Colony Optimization. There has been a lot of work in ant colony optimization. There are different versions of classification tasks based on the characteristics of attribute and class labels. All these works are done in several different classification methods in ant colony. The classification problems in ant colony optimizations are, Ant-Miner [1], cAnt-Miner, hAnt-Miner, Ant-tree Miner [2], hmAnt-Miner [3]. Here it takes inspiration from real ants to solve a problem in an optimal way. Here we can see an evolution of ant colony system. Shortest path is discovered using the pheromone trails. Basic AntMiner is used to extract classification rules in the form of IF <term1 AND term2 AND · · ·> THEN <class>. Here it consists of following steps: construction of rule, rule pruning, pheromone updation, normalization. Limitation of Ant-Miner is that it cannot cope up with continuous values. Ant-Miner [1] follows a sequential covering approach to find a list of classification rules covering almost all training cases. In a few respects Ant-Miner is quite different from other ant algorithms. This is the first ACO Algorithm for discovering classification rules. it classifies dataset of single class label. It can process only discrete values. This is a drawback of Ant-Miner. Construction graph is based on information theory. Which is calculated by using entropy. In [2], the main difference is the construction of rules that have less structural constraints with rules that are more complex. Another difference is that, no pruning has been used forthe rules. Thus, the consequent of each rule is determined after the construction of rule. The name Tree-Miner is adopted here for representation. In Tree-Miner, pheromones are associated with links in the graph. Cov-rate is a variable that is used here to control the speed at which the algorithm converges so that it can avoid the problem of immature convergence. hAnt-Miner, divides the rule construction process into two different ant colonies-creating antecedent of rules and creating consequent of rules both work in a cooperative fashion in order to discover the list of classification rules. A sequential covering algorithm is employed to cover all examples. Ants in the antecedent part create a rule on the antecedent graph and on consequent part create a rule on the consequent graph. In order to create a rule, Rulet from the antecedent part is paired with an ant from the consequent part and hence synchronizes 2 colonies. The only requirement is that both colonies should contain same number of ants. The hmAnt-Miner algorithm [3] is the multi label version of previous one. The proposed hmAnt-Miner algorithm is competitive with Clustured Hierarchical Multi label Classification the most accurate of the clustering algorithms in terms of both predictive accuracy and classification model size. Additionally, hmAnt-Miner outperformed single classification method (ClusSC) in terms of predictive accuracy on the combined (both FunCat and Gene Ontology) data sets and it has discovered a much simpler classification model than Clus-SC. hmAntMiner differ from h-AntMiner in certain aspects:

• The consequent of a rule is calculated using a deterministic procedure based on the examples covered by the rule, allowing the creation of rules that can predict more than one class label at the same time (multi-label rules). Therefore, hmAnt-Miner uses a single construction graph in order to create a rule-only the antecedent is represented in the construction graph;

• The heuristic function is based on the Euclidean distance, in which each example in the dataset is represented by a vector of class membership values in the Euclidean space. By using entropy, it is possible to take into account the relationship between class labels given that examples belonging to related (ancestor/descendant) class labels will be more similar than examples belonging to unrelated class labels. The use of the Euclidean distance was inspired by a similar use in the Clus-HMC algorithm for hierarchical multi label classification, which is based on the paradigm of decision tree induction, rather than rule induction. Note that the Euclidean distance is used as the heuristic information, as well as in the dynamic discretization procedure of continuous attributes;

• The rule quality is evaluated using a distance-based measure, which is a more suitable evaluation measure for hierarchical multi-label problems;

• The pruning procedure is not applied to the consequent of a rule. The consequent of a rule is (re-)calculated whenits antecedent is modified during pruning, since the set of covered examples might have changed. Real-world data from UCI repository is used to evaluate these systems [4]. This paper not only proposes a new entropy measure for measuring a node with hierarchical class labels,but a new strategy too to determine the most suitable concept label for a leaf node. In this project accuracy of the prediction is guaranteed without losing the precision. Finally transform the DT into if-condition-then-label rules by traversing each path of the tree from the root node to the leaf nodes. In [5], three approaches are proposed. In the first one, it defines an independent single-label classification task for each class (SC). The hierarchy introduces dependency among the classes. While hierarchical property was ignored by the first approach, it was exploited by the second approach i.e, hierarchical single label classification (HSC). Classes may have multiple parents (DAG structure). 24 yeast datasets are used for comparing these approaches. Many real-world applications involve multi label classification, in which the labels are organized in the form of a tree which is having a hierarchical structure. A novel approach is proposed in [6], which can be used on both tree- and DAG-structured hierarchies. Using a simple greedy strategy, the proposed algorithm is computationally efficient, easy to implement, does not suffer from the problem of insufficient training data in classifier training, and can be readily used on large hierarchies. Much work in hierarchical multi-label classification (HMC) has been motivated by text classification.[7] consists mostly of Bayesian and kernel-based classifiers. On this, it is a decision tree-based approach related to predictive clustering trees. They start from a different definition of variance and then kernalize this variance function and the result is a decision tree induction system that can be applied to structured output prediction using a method similar to the large margin methods. Therefore, this method could also be used for HMC after defining a suitable kernel. In another research [8] a hierarchical text classification problem setting is done where each text document belongs to one and only one class at the bottom level of a topic hierarchy. For each topic in an internal node of the hierarchy, a Bayesian classifier is learned that distinguishes between the possible subtopics, using only those training instances that belong to the parent topic. Test documents are then classified by filtering them through the hierarchy, predicting one topic at each level, until the documents reach the bottom level, thereby ensuring the hierarchy constraint. Errors made at higher levels of the hierarchy are unrecoverable at the lower levels. The procedure is similar to the HSC approach. Here the hierarchical approach compares favorably with the simple approach of constructing a single large classifier over a flattened topic space. In this case, feature selection phase plays a crucial role in the performance of the resulting classifier. The hierarchical classification scheme begins by applying probabilistic feature selection to the entire training dataset. The resulting reduced feature set is then used to build a probabilistic classifier of the hierarchy. [9] proposes a method for multi-label classification in the context of functional genomics. Here, a tree predicts not a single class but a vector of Boolean class variables. Also proposed a simple adaptation of C4.5 which normally uses class entropy for choosing the best split, their version uses the sum of the entropy of the class variables. They have extended the method to predict the classes on several levels of the hierarchy, assigning a larger cost to misclassification higher up in the hierarchy, and presented an evaluation on the 12 data sets. [10] presented a two-step approach where support vector machines (SVMs) are learned for each class separately, and then combined using a Bayesian network model so that the predictions are consistent with the hierarchy constraint. Progress has been made on multi-label classification on treestructured hierarchies. A simple remedy is to allow a classifier for a particular node to predict positive only if the classifier of its parent also predicts positive. However, most existing multi-label classification algorithms do not take the label structure into consideration. Instead, the labels are simply treated separately, leading to the need to train a large number of classifiers (one for each label). Moreover, as some labels (such as those at the lower levels of the hierarchy) may have very few positive examples, the training data become highly skewed, which can be problematic to many classifiers. Besides, the inconsistent labeling between child and parent causes difficulty in interpretation. Finally, the prediction performance is impaired as structural dependencies among labels are not utilized in the learning process. [4] presents classification of data using decision tree. Entropy is used as heuristic measure. In the case of decision tree after applying the algorithm to the input training data set, we can build a decision tree DT: T(V, E) where E is the set of edges and v is set of nodes. DT is as shown in figure 2. In the tree, the leaves are the final results of the concept labels. By this tree, we can predict customers interests. For example, if there is a female customer with an income of Rs.1324 the tree indicates that she will be interested in a <AppleLaptop>. Finally, transform the DT into a set of rules equivalent to the tree. Each rule is generated by traversing the decision tree, starting from the root up to the leaf nodes. Each path results in an "if-condition-then-label". The HLC (Hierarchical class Label Classifier) algorithm is designed to construct a DT from data with

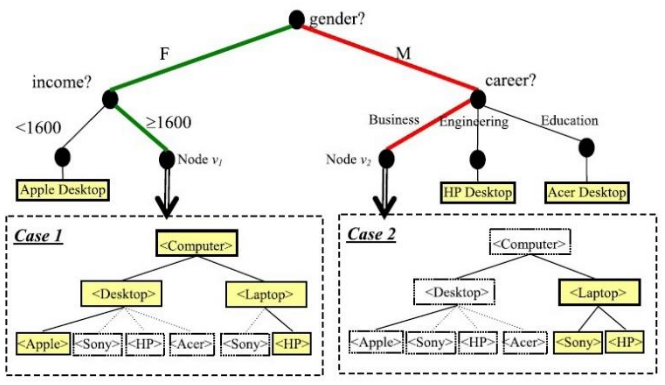


Fig 2. DT with hierarchical Structure of class labels

hierarchical class labels. It follows the standard framework of classical DT induction methods, such as ID3 and C4.5. While constructing the decision tree, the distribution of the labels of the data over the class hierarchical tree is considered; since the algorithm is designed to construct a DT with hierarchical class labels, therefore, the proposed measure should be capable of dealing with hierarchical class labels. In order to overcome the weakness of entropy-based method, here introduces a new measure, called hierarchical entropy value, by modifying the traditional entropy measure. It can help to measure the appropriateness of a node with respect tothe given class hierarchical tree. First, we need to define what the hierarchical information gain is, and then use it to develop a method for choosing the best splitting attributes. Select an attribute with the highest Hierarchical information gain (Hinfo Gain) value, which is chosen as the next test attribute for the current node. Shannon theory is used in standard top-down decision trees. Shannon defines a formal measure of entropy [12].

$$s = -\sum_{i=1}^{n} p_i log_2 p_i$$

where p_i is the probability of occurrence of an event (feature value) x_i being an element of the event (feature) X that can take values $x_1...x_n$. The Shannon entropy is a decreasing function of a scattering of random variable, and is maximal when all the outcomes are equally likely. Shannon entropy is the limiting case of Renyi entropy. Here we use Renyi entropy and Shannon entropy for comparative processes. This approach may be used in any decision tree and information selection algorithm. Here it compares the performance for each artificial dataset. Over the last decade or so, the need for systems capable of automated categorization has become more and more urgent. The main reason is that, with the advent of the world-wide web, the amount of available data is growing so fast that it is impractical to do it all manually.

3 Proposed System

Ant Colony Optimization is a system used for solving optimization problem. In the case of Ant colony Optimization algorithm, each path followed by an ant corresponds to a candidate solution. The amount of pheromone deposited by an ant is proportional to the quality of the candidate solution for choosing for the target problem. If there is more than one path through which they can pass. where W is the class attribute whose domain consists of the classes to be predicted. c is the number of classes. P $(w \mid A_i = B_{ij})$ is the probability of observing class w conditional on having observed $A_i = B_{ij}$. A is attribute and Bis possible values of each attribute.Therefore, the proposed normalized, information-theoretic heuristic function is: where η_{ij} is heuristic function. Here we calculate heuristic value for each node. Using this calculated value calculate probabilities of each each. Here we determine which node is to be used for generating rule.

Algorithm 1 High level pseudo code

```
1:  TrainingSet = $\{all\ training\ cases\}$
2:  RuleList = []; /*rule list is initialized with empty set*/
3:  typeconversion();
4:  while $(TrainingSet > Max\_uncovered\_cases)$ do
5:      $rule_{best} \leftarrow 0$;
6:      $i \leftarrow 1$;
7:      if data set contains numeric values then
8:          $calculateAttrbuteStat()$;
9:      end if
10:     initialize all trails with same values for all pheromone trails;
11:     repeat
12:         $rule_{current} \leftarrow 0$;
13:         for $j \leftarrow 1$ $to\ colony\_size$ dodo
14:             calculateEntropy();
15:             calculateHeuristicFunction();
16:             calculateProbabilities();
17:             Determine rule consequent;
18:             Multiply heuristic information with pheromone value and find the maximum value among them and choose the corresponding attribute as best one
19:             pruneRule()
20:             updatepheromone using equation 6
21:             $(t \geq No\_of\_ants)\,OR\,(j \geq No\_rules\_converged)$
22:             choose best rule among all rules constructed by the ant
23:             Add best rule to Rulelist
24:             Trainingset = TrainingSet
                    $-\ \{set\ of\ cases\ covered\ by\ best\ rule\}$
25:
```

Algorithm 2 typeconversion

```
1:  input all data to be convert
2:  if data contains "{" then /*checking set of values*/
3:      if class label contains hierarchical structure then
4:          find combinations of each node with other nodes except its parent and grand parent node and add to attribute_list
5:      else
6:          $generate\ combinations\ for\ all\ values$ and add to attribute_list
7:      end if
8:  else
9:      add all values to attribute_list
10: end if
11: Find index values of each =0
```

Algorithm 3 calculateEntropy

```
1: Take all examples from the dataset
2: Calculate entropy using equation 2
3: Store all these values of nodes
```

Algorithm 4 calculateHeuristicFunction

```
1: Take all entropy values obtained after calculation
2: Calculate heuristic value using equation 3
3: Store heuristic values of each nodes
```

From these probability values nodes with higher probabilityis taken for rule genearation.

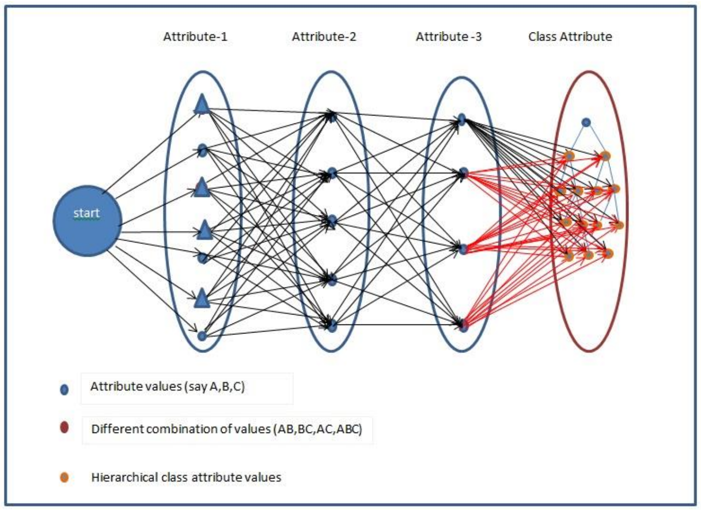
Algorithm 6 Prunerule

```
1: for EACH term in the rule current to be pruned do
2:     Temporarily remove term and assign to the rule consequent the most frequent class among the examples covered by the rule antecedent;
3:     Evaluate rule quality;
4:     Reinstate term t in rule antecedent;
5: end for
6: if rule quality was improved w.r.t. original rules Quality in some iteration of the FOR loop then
7:     Remove permanently the term whose removal improves rule current most;
8: end if
```

From figure 2,there are several nodes with internal nodes. Each node represents attributes and internal node represents the possible values of the attribute. In this structure, sub nodes of the first node represented by using a circle and a square. Circle is for attribute values and another for possible combinations of attribute values. Each sub node is connected to each other sub nodes in the other nodes. When an ant starts from the start node it can take any path to reach into the target node. Class attribute is hierarchical in structure.Since it is multi labeled,

Fig 3. Structure of our Ant Colony Optimization System

nodes from non class attribute is connected to all other nodes in the hierarchical tree. But it is not possible to have a


Attribute-1
Attribute-2
Attribute-3
Class Attribute
start
Attribute values (say A,B,C)
Different combination of values (AB,BC,AC,ABC)
Hierarchical class attribute values

connection with a parent node and its own child node at the same point of time. Instead, it can have two different values with two different parent nodes. In this scenario, node labeled class attribute is the target node. Maximum number of ants which traverse through the optimal path gives an optimal solution for the problem. The path(s) with larger amount of pheromone has higher probability of being chosen by the ant. The design of the ACO consists of:

- Constructing a rule
- Heuristic information
- Updating pheromone
- Transition function

Description of our proposed system can be divided into five major parts:

A. General Description

Our goal is to extract a rulein the form of IF <term1 AND term2 AND · ·> THEN <subset of labels> Here each term is a triplet consisting of hattribute, operator, value$_i$. A training set consists of all the set of training cases. At first, the discovered rule set consist of empty rules. On each iteration a classifier rule is discovered and added to the discovered rule set. The training cases that are correctly covered by the rule are removed from the training set. This process is repeated until the number of uncovered training cases is greater than a user specified threshold. From the figure, circle representsthe attribute values and triangle represents the different possible combinations of attribute values of the corresponding attribute. First, Ant starts with an empty rule set. One term is added to the set forms a partial rule set. Current partial rule corresponds to the current partial path followed by that ant. The choice of a term to be added to the current partial rule corresponds to the choice of the direction in which the current path will be extended. The choice of the term to be added to the current partial rule depends on both a problem-dependent heuristic function and on the amount of pheromone associated with each term, as will be discussed in detail in the next subsections. Ant keeps adding one-term-at-a-time to its current partial rule until one of the following two stopping criteria is met:

Any term to be added to the rule would make the rule cover a number of cases smaller than a user-specified threshold, called Min cases per rule (minimum number of cases covered per rule). All attributes have already been used by the ant, so that there are no more attributes to be added to the rule antecedent. Note that each attribute can occur only once in each rule, to avoid invalid rules. Rule constructed by Ant is pruned in order to remove irrelevant terms. Now we only mention that these irrelevant terms may have been included in the rule due to stochastic variations in the term selection procedure and/or due to the use of a shortsighted, local heuristic function - which considers only one-attribute-at-a-time, ignoring attribute interactions. The amount of pheromone in each trail is updated, increasing the pheromone in the trail followed by Ant (according to the quality of rule) and decreasing the pheromone in the other trails (simulating the pheromone evaporation). Then another ant starts to construct its rule, using the new amounts of pheromone to guide its search. This process is repeated until one of the following two conditions is met:

- The number of constructed rules is equal to or greater than the user-specified threshold No_of_ants.
- The current Ant has constructed a rule that is exactly the same as the rule constructed by the previous No_rules_converg - 1 ants, where No_rules_converg stands for the number of rules used to test convergence of the ants.

B. Heuristic Function

For each term term$_{ij}$ that can be added to the current rule, it computes heuristic function that is an estimate of the quality of this term, with respect to its ability to improve the predictive accuracy of the rule. Here heuristic function is based on the Euclidean Distance of the class label where each example is represented by a vector of class membership values in the Euclidean space. Here, the heuristic information of a term corresponds to the variance of the set of examples covered by the term. In order to calculate the variance, the class labels of each example are represented by an index value which is unique for each one.Since dataset consists of set valued attribute, we need to find the possible combinations of these values.The value of ηij for termij involves a measure of the amount of information associated with that term. For each termij of the form $A_i = a_{ij}$ where A_i is the ith attribute and B_{ij} is the j th value belonging to the domain of A_i its entropy is given by,

entropy = $H\ (W\,|\,A_i = B_{ij})$ (1)

$= -\sum_{w=1}^{c} p(W|A_i = B_{ij)}\, log_2 p(W|A_i = B_{ij})$(2)

where W is the class attribute whose domain consists of the classes to be predicted. c is the number of classes. P $(w\,|\,A_i = B_{ij})$ is the probability of observing class w conditional on having observed Ai = Bij. Therefore, the proposed normalized, information-theoretic heuristic function is:

$$\eta_{ij} = \frac{log_2 c - H\left(W|A_i = B_{ij}\right)}{\sum_{i=1}^{a}\sum_{j=1}^{b} log_2 c - H\left(W|A_i = B_{ij}\right)} \qquad (3)$$

where A is the number of attributes and B is the possible values of corresponding attribute.In the case of set valued attribute the possible combinations of values also need to be considered. After calculating heuristic function, it is essential to find the probabilities of each node to determine which node(attributevalue pair) should be taken for generating rule. probabilities can be calculated by using equation:

$$P_{ij} = \frac{\eta_{ij}.\tau ij\,(t)}{\sum_{i=1}^{a}\sum_{j=1}^{b}\eta_{ij}.\tau ij\,(t)} \qquad (4)$$

where P_{ij} is probabilities of each values of curresponding attribute.

C. Rule pruning

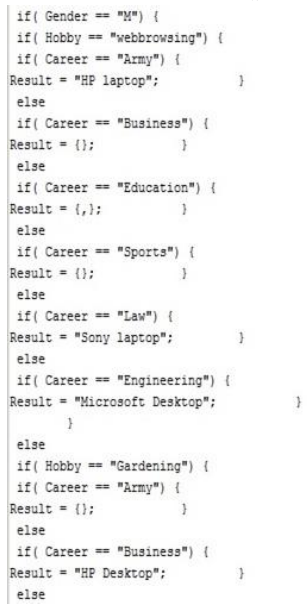
Rule pruning is a technique commonly used in data mining. As mentioned earlier, the main goal of rule pruning is to remove irrelevant terms in the rule. And hence it increases the power of prediction of the rule. It helps to avoid its over fitting to the training data and another motivation for rule pruning is its capability to improve the simplicity of the rule, since a shorter rule can easily be understood by the user than a longer one. The basic idea is to iteratively remove one and only one term at a time from the rule while this process improves the quality of the rule. More precisely, during first iteration one starts with the full rule. Then remove each of the terms of the rule at a time and the quality of the resulting rule is computed using a given rule-quality function.The term whose removal improves the quality of the rule is effectively removed from it, completing the first iteration. In the next epoch the term is removed whose removal improves the quality of the iteration and so on. This process is repeated until the rule has just one term or until there is no term whose removal will improve the quality of the rule. The Quality of the rule is defined by,

$$Q = \frac{\sum TP}{\sum TP + FN} * \frac{\sum TN}{\sum TN + FP} \qquad (5)$$

where TP is True Positive which is defined as the number of items correctly classified as positive class TN is True Negative which are items correctly classified as negative class FP is False Positive which are items incorrectly classified as positive class FN is False Negative which are items incorrectly classified as negative class it can be written as
Sensitivity * Specificity.
Sensitivity is the ratio of number of true positives to number of positives and specificity is the ratio of number of negatives to number of negatives.

D. Pheromone Updation

The amount of pheromone deposited initially at each path is inversely proportional to the total number of values of all attributes. Pheromone updation for a termij is performed based



```
 if( Gender == "M") {
 if( Hobby == "webbrowsing") {
 if( Career == "Army") {
Result = "HP laptop";          }
 else
 if( Career == "Business") {
Result = {};          }
 else
 if( Career == "Education") {
Result = {,};          }
 else
 if( Career == "Sports") {
Result = {};          }
 else
 if( Career == "Law") {
Result = "Sony laptop";          }
 else
 if( Career == "Engineering") {
Result = "Microsoft Desktop";          }
        }
 else
 if( Hobby == "Gardening") {
 if( Career == "Army") {
Result = {};          }
 else
 if( Career == "Business") {
Result = "HP Desktop";          }
 else
```

on

Fig 4. Rules Generated using Decision tree

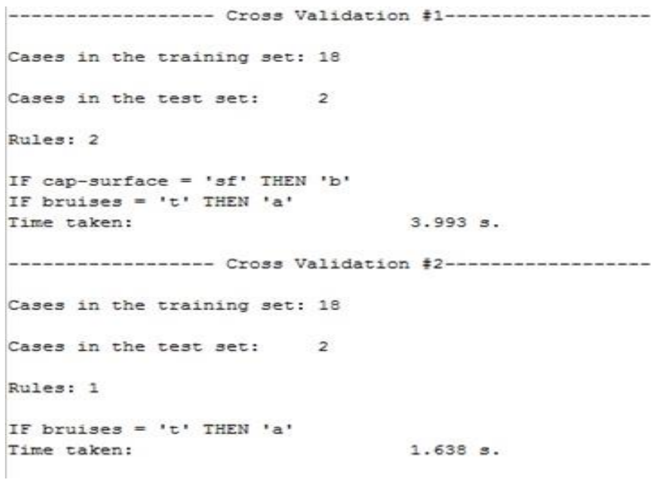
the term below:

$\tau_{ij}(t + 1) = \tau_{ij}(t) + \tau_{ij}(t).Q$ (6)

Where i,j are element of a set of terms occurring in the rule constructed by the ant at iteration t ,Decreasing the amount of pheromone associated with each term is done by normalizing the value of each pheromone. It is done by dividing the value of each pheromone by the summation of all pheromone, which will result in the reduction of the normalized amount of pheromone for each unused term.

4 Experimental Results

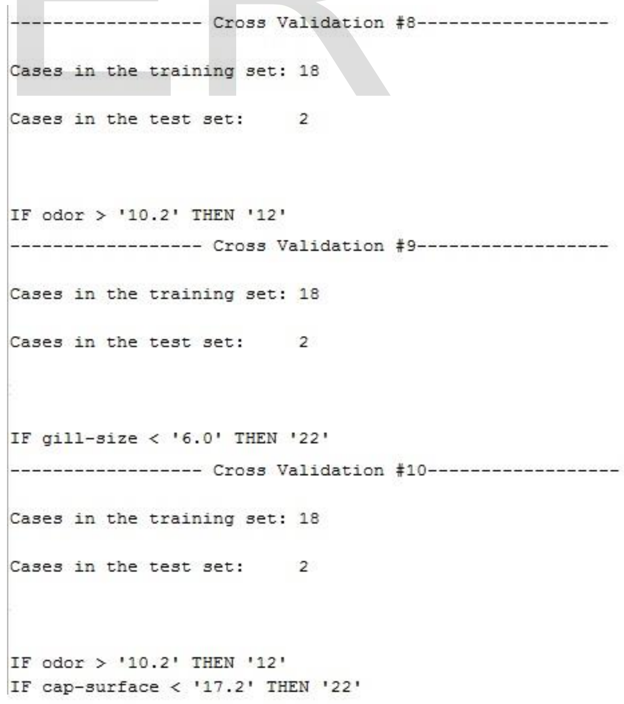
Here we are assessing empirically against (1) the state-ofthe-art decision tree algorithm for hierarchical classification using Renyi entropy(CRE)(2) an approach consisting of training hier



```
------------------ Cross Validation #1------------------

Cases in the training set: 18

Cases in the test set:     2

Rules: 2

IF cap-surface = 'sf' THEN 'b'
IF bruises = 't' THEN 'a'
Time taken:                        3.993 s.

------------------ Cross Validation #2------------------

Cases in the training set: 18

Cases in the test set:     2

Rules: 1

IF bruises = 't' THEN 'a'
Time taken:                        1.638 s.
```

Fig 5. Rules generated by our algorithm

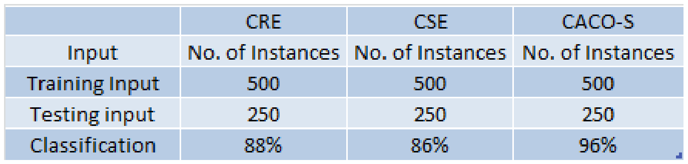
archical class label using shannon entropy(CSE).(3)our basic single label classification method using Ant Colony Optimiza



```
------------------ Cross Validation #8------------------

Cases in the training set: 18

Cases in the test set:     2



IF odor > '10.2' THEN '12'
------------------ Cross Validation #9------------------

Cases in the training set: 18

Cases in the test set:     2



IF gill-size < '6.0' THEN '22'
------------------ Cross Validation #10------------------

Cases in the training set: 18

Cases in the test set:     2



IF odor > '10.2' THEN '12'
IF cap-surface < '17.2' THEN '22'
```

Fig 6. Rules generated by our algorithm for continuous values

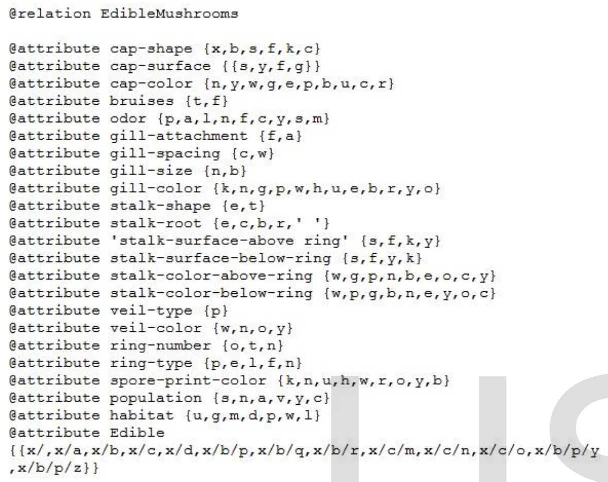
tion(CACO-S). The dataset in the experiments consisting of synthetic data. We have taken 500 instances for training and 250 instances for testing each method. And we evaluated the performance for each one. Details of data set that we used for testing of classification is given in figure 7. The assessment of the classification algorithms in this scenario in terms of accuracy was performed using the evaluation measures for hierar-



	CRE	CSE	CACO-S
Input	No. of Instances	No. of Instances	No. of Instances
Training Input	500	500	500
Testing input	250	250	250
Classification	88%	86%	96%

chical classification. further more, in the case of hierarchical
Fig 7. Input data details of different methods

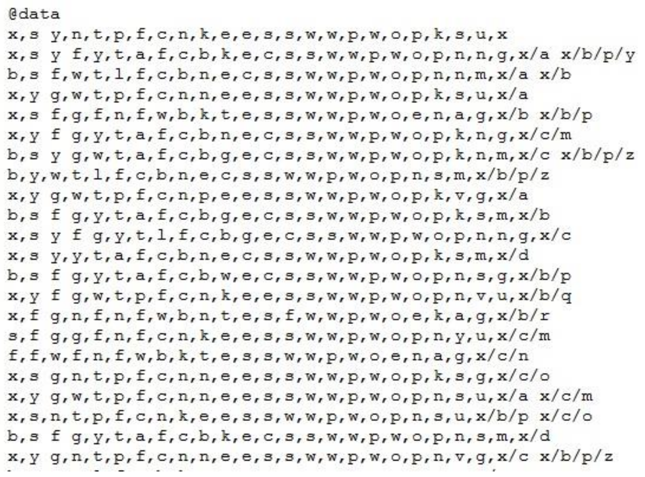
multilabel experiments also the classification algorithm is evaluated. Some rules found by our algorithm are given in



```
@relation EdibleMushrooms

@attribute cap-shape {x,b,s,f,k,c}
@attribute cap-surface {{s,y,f,g}}
@attribute cap-color {n,y,w,g,e,p,b,u,c,r}
@attribute bruises {t,f}
@attribute odor {p,a,l,n,f,c,y,s,m}
@attribute gill-attachment {f,a}
@attribute gill-spacing {c,w}
@attribute gill-size {n,b}
@attribute gill-color {k,n,g,p,w,h,u,e,b,r,y,o}
@attribute stalk-shape {e,t}
@attribute stalk-root {e,c,b,r,' '}
@attribute 'stalk-surface-above ring' {s,f,k,y}
@attribute stalk-surface-below-ring {s,f,y,k}
@attribute stalk-color-above-ring {w,g,p,n,b,e,o,c,y}
@attribute stalk-color-below-ring {w,p,g,b,n,e,y,o,c}
@attribute veil-type {p}
@attribute veil-color {w,n,o,y}
@attribute ring-number {o,t,n}
@attribute ring-type {p,e,l,f,n}
@attribute spore-print-color {k,n,u,h,w,r,o,y,b}
@attribute population {s,n,a,v,y,c}
@attribute habitat {u,g,m,d,p,w,l}
@attribute Edible
{{x/,x/a,x/b,x/c,x/d,x/b/p,x/b/q,x/b/r,x/c/m,x/c/n,x/c/o,x/b/p/y
,x/b/p/z}}
```

figure:
Fig 9. Header information of ARFF file

5. The sample of rules generated for a continuous dataset is given in figure 6. Here we used ARFF (Attribute Relation File Format) data files for the empirical study which consists of two sectionsHeader and Data information.Header contains



```
@data
x,s y,n,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,k,s,u,x
x,s y f,y,t,a,f,c,b,k,e,c,s,s,w,w,p,w,o,p,n,n,g,x/a x/b/p/y
b,s f,w,t,l,f,c,b,n,e,c,s,s,w,w,p,w,o,p,n,n,m,x/a x/b
x,y g,w,t,p,f,c,n,n,e,e,s,s,w,w,p,w,o,p,k,s,u,x/a
x,s f,g,f,n,f,w,b,k,t,e,s,s,w,w,p,w,o,e,n,a,g,x/b x/b/p
x,y f g,y,t,a,f,c,b,n,e,c,s,s,w,w,p,w,o,p,k,n,g,x/c/m
b,s y g,w,t,a,f,c,b,g,e,c,s,s,w,w,p,w,o,p,k,n,m,x/c x/b/p/z
b,y,w,t,l,f,c,b,n,e,c,s,s,w,w,p,w,o,p,n,s,m,x/b/p/z
x,y g,w,t,p,f,c,n,p,e,e,s,s,w,w,p,w,o,p,k,v,g,x/a
b,s f g,y,t,a,f,c,b,g,e,c,s,s,w,w,p,w,o,p,k,s,m,x/b
x,s y f g,y,t,l,f,c,b,g,e,c,s,s,w,w,p,w,o,p,n,n,g,x/c
x,s y,y,t,a,f,c,b,n,e,c,s,s,w,w,p,w,o,p,k,s,m,x/d
b,s f g,y,t,a,f,c,b,w,e,c,s,s,w,w,p,w,o,p,n,s,g,x/b/p
x,y f g,w,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,n,v,u,x/b/q
x,f g,n,f,n,f,w,b,n,t,e,s,f,w,w,p,w,o,e,k,a,g,x/b/r
s,f g,g,f,n,f,c,n,k,e,e,s,s,w,w,p,w,o,p,n,y,u,x/c/m
f,f,w,f,n,f,w,b,k,t,e,s,s,w,w,p,w,o,e,n,a,g,x/c/n
x,s g,n,t,p,f,c,n,n,e,e,s,s,w,w,p,w,o,p,k,s,g,x/c/o
x,y g,w,t,p,f,c,n,n,e,e,s,s,w,w,p,w,o,p,n,s,u,x/a x/c/m
x,s,n,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,n,s,u,x/b/p x/c/o
b,s f g,y,t,a,f,c,b,k,e,c,s,s,w,w,p,w,o,p,n,s,m,x/d
x,y g,n,t,p,f,c,n,n,e,e,s,s,w,w,p,w,o,p,n,v,g,x/c x/b/p/z
```

name of the relation,attribute list and its type. Data contains as
Fig 10. Structure of data in dataset in ARFF file
name indicates all data in the training or testing dataset. Header looks like figure 9. We represent those attributes which is having a set of values with two curly brackets in order to identify that it is a set valued attribute. If an attribute has a set of values then we represent it by using an extra bracket. Dataset may contain numerical data. There are two categories of numerical data: discrete and continuous. If there are only finite number of possible values for an attribute the type of data is calleddiscrete data. It is
usually occurring in a case of certain number of values. Data is said to be continuous if and only if the values belong to it may take an interval. Height, weight, temperature etc. are examples of continuous data. Here continuous data is represented using another set of brackets in order to identify the continuous data. There are two categories: data represented using closed intervals and discrete values. In the case of closed interval values in the dataset is represented by writing in between "(" and ")" brackets. Discrete values are represented by writing relation in between"(" and")". The average of each attribute values are

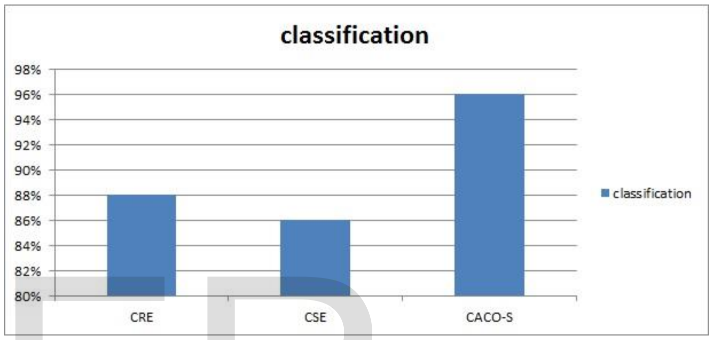


Fig 8. Accuracy(%) graph for different algorithms

taken for generating rule. Last attribute is class attribute which is arranged in a hierarchy. As it is also a multi labeled data, here also we used double curly brackets to represent the same. There is a hierarchical relation in the case of class labels. "=" is used to represent the parent child relation. Set of values in dataset is separated by using white space. In the case of continuous val

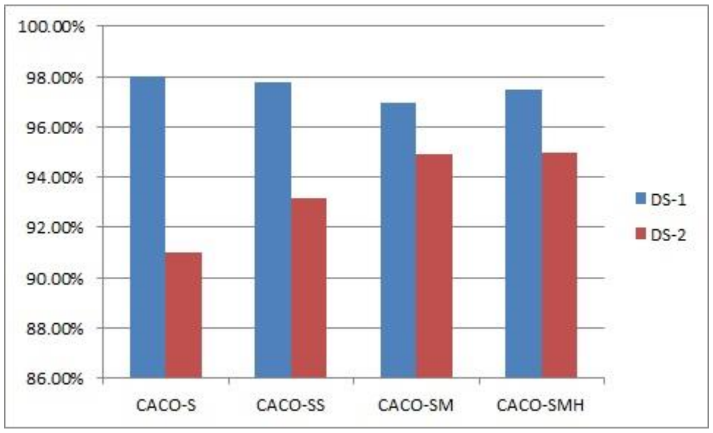


Fig. 11. Classification Accuracy(%) graph for differentACO algorithms

ues we use "<" and ">" like symbols while generating rules. If attribute values are greater than mean, then use ">" else we use "<" during rule generation. Figure 10 shows the structure of data in dataset in ARFF file. Each attribute value in the dataset is separated by using comma.Set of values for an attribute in the dataset is separated by using a whitespace. In order to develop a system, it was necessary to build a dataset which con

sists of all thesecharacteristics. We built an artificial dataset with set valued attribute and class labels are hierarchical and multi labeled as mentioned earlier. Also, we use two datasets from UCI repository. We modified our dataset for our convenience. Here it consists of 500 instances with 22 attributesincluding one attribute with set of values. In the zoo dataset there are 101 instances. The learning task of this dataset is to predict the type of animal. For the evaluation we used two modified dataset DS-1 and DS-2. Class label attribute also contains multiple values. Details of dataset is given in Table 1. We have tested for single label classification,set valued single label classification, set valued multi label classification,set valued multi label hierarchical classification. Accuracy of each one in each dataset is given below. All experiments wereconducted running a 10-fold cross validation process, in which break dataset into 10 equal set of size. Train 9 among them and test the remaining one. Repeat this process 10 times and find the mean accuracy of it. Accuracy rate of datasets using different methods is given in Table 2. The numbers after " + / −" denote standard deviations. Figure 11 represents the accuracy graph of different ACO techniques.

TABLE I
SUMMARY OF DATASET USED IN OUR EXPERIMENTS

Dataset	Number of Instances	Number of Attributes
DS-1	8124	23
DS-2	101	17

TABLE II
CLASSIFICATION ACCURACY(%) FOR DIFFERENT ACO ALGORITHMS USING 10-FOLD CROSS VALIDATION

Dataset	Dataset Type	Accuracy Rate on Testing
DS-1	CACO-S	97.99% +/- 0.27%
DS-2	CACO-S	87.41% +/- 3.57%
DS-1	CACO-SS	97.76% +/- 0.59%
DS-2	CACO-SS	90.39% +/- 2.74%
DS-1	CACO-SM	96.93% +/- 0.42%
DS-2	CACO-SM	92.24% +/- 2.65%
DS-1	CACO-SMH	97.48% +/- 0.69%
DS-2	CACO-SMH	92.99% +/- 2.57%

5 Conclusion and Future work

Our work has proposed an algorithm for rule discovery in the case of set valued attribute with hierarchical multi label class labels. The goal of our system is to discover classification rules in data sets. The algorithm is based both on research on the general behavior of real ant colonies and on data mining concepts and principles. We have compared the performance of our system and the well-known Renyi algorithm using synthetic datasets. Dataset may also contain continuous data. Here we also dealt with continuous data as well. This work can be extended in two ways as follows: class label may contain fuzzy values, so by considering the membership of samples in all possible fuzzy sets, it should be possible to find the fuzzy rules. Fuzzy rule is a tool for expressing pieces of knowledge in fuzzy logic. Fuzzy rules are rules whose antecedents, consequences or both are fuzzy rather than crisp. Fuzzy rules can be of "if x = A then y is B" where A and/or B might be fuzzy. Certainty rules with fuzzy condition parts enable us to relate typicality and certainty, by interpreting "the more x is A" as "the more typical x is", as in the example "the more typical bird, the more certain it flies" In such a case, typicality can be appreciated in terms of the weight of the bird, the length of the wings and so on. So, such a rule requires rather precise information about the considered bird in order to get a useful conclusion. We can apply our method in such problems too. Alternative approaches like genetic algorithm can also be adopted to derive a solution to do this problem.

Acknowledgment

We thank all our colleagues and friends for giving us assistance in making this research work successful.

References

[1] Rafael S. Parpinelli1, Heitor S. Lopes1, and Alex A. Freitas2 ,"' Data Mining with an Ant Colony Optimization Algorithm"'IEEE Transactions on Evolutionary Computation,(2002)
[2] Negar ZakeriNejad, Amir H, Bakhtiary and MortezaAnaloui, "Classification Using Unstructured Rules and Ant Colony Optimization", Proceedings of the International Multi Conference of Engineers and Computer Scientists 2008 Vol I IMECS 2008, 19-21 March, 2008, Hong Kong
[3] Fernando E. B. Otero, Alex A. Freitas, Colin G. Johnson,"'A Hierarchical Multi-Label Classification Ant Colony Algorithm for Protein Function Prediction"', Memetic Comp. (2010)2:165181DOI 10.1007/s12293-010- 0045-4
[4] Yen-Lian Chen, Hsiao-Wei Hu, KweiTang,"'Constructing a decision Tree from data with hierarchical class labels"', Expert Systems with Applications (2009)
[5] Blockeel, H., Schietgat, L., Struyf, J., Dzeroski, S., and Clare, A. 2006.," 'Decision trees for hierarchical multi-label classification: A case study in functional genomics"', Knowledge Discovery in Databases: PKDD 2006, Proceedings 4213, 1829
[6] Wei Bi, JamesT. Kwok,"Multi-Label Classification on Tree and DAGStructured Hierarchies" Proceedings of the 28th international conference on machine learning(2011)
[7] Rousu, J., Saunders, C., Szedmak, S., and Shawe-Taylor, J. ,"Kernelbased learning of hierarchical multi-labelclassification models", Journal of MachineLearning Research, 7:1601.1626, 2006

[8] Koller, D. and Sahami, M. 1997,"Hierarchically classifying documents using very few words", In Proceedings of ICML-97, 14th International Conferenceon Machine Learning, D. Fisher, Ed. Morgan Kaufmann Publishers, SanFrancisco, US, 170178

[9] Clare, A. and King, R. D. (2001),"'Knowledge discovery in multi-label phenotype data"', In 5th European conference on principles of data mining and knowledge discovery (pp. 4253)

[10] Barutcuoglu, Z. and Troyanskaya, O.G," Hierarchicalmulti-label prediction of gene function"', Bioinformatics.,22(7), 2006

[11] Vens, C., Struyf, J., Schietgat, L., Dzeroski, S., and Blockeel,"Decision trees for hierarchical multi-label classification"', Machine Learning 73, 2, 185214

[12] Tomasz Maszczyk and Wlodzis law Duch," Comparison of Shannon, Renyi and Tsallis Entropy used in Decision Trees"', Lecture Notes in Computer Science, Vol.5097 643-651,2008

[13] Allen Chan and Alex A. Freitas,"' A New Ant Colony Algorithm for Multi-Label Classification with Applications in Bioinformatics"', In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006), pp. 27-34

[14] Sonal P. Rami,Mahesh H. Panchal,"' Comparative Analysis of Variations of Ant-Miner by Varying Input Parameters, International Journal of Computer Applications (0975 8887) Volume 60 No.3, December 2012

[15] Fernando E. B. Otero, Alex A. Freitas, and Colin G. Johnson,"'A New Sequential Covering Strategy for Inducing Classification Rules with Ant Colony Algorithms"', IEEE Transactions on Evolutionary Computation Vol. 17, No. 1, February 2013