Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 5, May 2015, pg.457 – 462

RESEARCH ARTICLE

Applying Ant Colony Optimization for Inducing Decision Tree from Mixed Type of Relational Multiclass Data

[1]PRIYA D. PATEL, [2]PROF. MAHESH PANCHAL

[1]Student, M.E., Dept. Of Computer Engineering, Kalol Institute of Technology and Research centre, Kalol-382721, Gujarat.

[2]Deputy Director, Gujarat Technology University, Ahmadabad, Gujarat

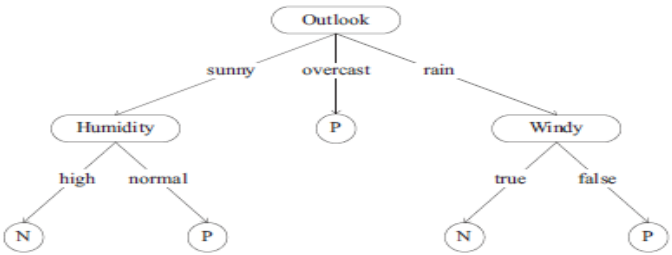
gopi8890@gmail.com

ABSTRACT: Decision trees have been widely used in data mining and machine learning as a comprehensible knowledge representation. Originally the method to construct decision tree follows greedy approach which results in local tree and classification rules. Ant Colony Optimization (ACO) is a meta-heuristic approach used to get more optimal solution compared to other methods from large search space. In proposed work we can create a decision tree using Discretization and Pruning method. So we reduce the size of the tree and improve the predictive accuracy.

Keywords— decision tree, ant colony optimization, Ant tree miner, Discretization, pruning, proposed method, experiment results, conclusion.

I. Introduction

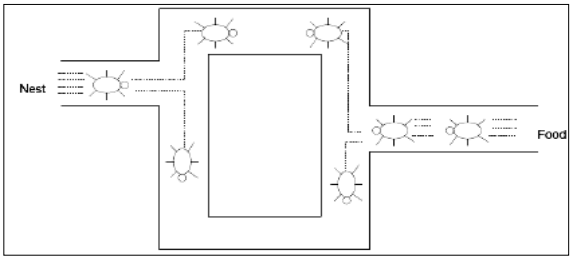
The Decision tree induction is the learning of decision from class label training tuple. A decision tree is flow charts like a tree structure where each internal node (non leaf node) denotes a test on attributes. Each branch represents an outcome of the test and each node leaf node (or terminal node) holds class label. The top most nodes are the root node. Decision trees provide a comprehensible graphical representation of a classification Model, where the internal nodes correspond to attribute tests (decision nodes) and leaf Nodes correspond to the predicted class labels—illustrated in Fig. 1.1.



[Figure 1 : Decision tree[1]]

ANT COLONY OPTIMIZATION

Ant colony optimization is a population based search technique for the solution of combinatorial optimization problems which is inspired by this behaviour. The ant finds the shortest path between their nest and a food source using pheromone tails.



[Figure 2: ACO based shortest path]

II. Ant Colony Optimization Procedure [1]

There are four main procedures involved
Input: problem's construction graph
Output: best solution

1. Initialize ();
2. While termination condition not met do
   ConstructAntSolutions ();
   ApplyLocalSearch ();
   Update Pheromones ();
3. End while
4. Return best solution.

1. Initialize: In This procedure initializes the Amount of pheromone (represented as a Pheromone matrix) and heuristic Information associated with the vertices or edges of the Construction graph.
2. ConstructAntSolutions: In this procedure incrementally builds candidate solutions By creating paths on the problem's construction graph, simulating the movement of an artificial ant.
3. ApplyLocalSearch: This procedure is an used to a further refine a solution Created By an ant. In that a local search algorithms is introduce to a small modifications of a Solution in order to explore neighbour solutions.
4. Update Pheromones: In this procedure updates the pheromone associated with the Components of – vertices or edges of the problem's construction graph . In that by Either increasing the amount of pheromone or decreasing the amount of pheromone Components used to create good solutions have their pheromone increased, while if Components not used will have their pheromone gradually decreased.

III. Different Ant Colony Algorithm

ANT TREE MINER [1]

In the Ant-Tree-Miner algorithm follows the traditional structure of ACO algorithms. First of all it starts by initializing the pheromone values and computing the heuristic information for each attribute of the training set. Then, it enters in an iterative loop (while loop) where each ant in the colony creates a new decision tree.
An ant creates a decision tree in a top-down fashion by probabilistically selecting attributes to be added as decision nodes based on the amount of pheromone and heuristic information. The decision tree create tree procedure takes three parameters, the set of training examples, the set of predictor attributes and the edge being followed by the ant, which at the start of the procedure corresponds to the default edge('_').

Once the tree construction procedure has finished, the created tree is pruned in order to simplify the tree and thus potentially avoid over fitting of the model to the training Data. Over fitting is the phenomenon where the model is too adjusted to the training Data and as a result does not have a good predictive accuracy on the test set, unseen During training.

After that pruning method the tree is evaluated and the iteration-best tree (*treeib*) is updated. if newly created tree is greater than the quality of the current iteration - best tree. Finally, the iteration-best tree (*treeib*) constructed by the ants is used to update the pheromone values, the global-best tree (*treegb*) is stored or updated and a new iteration of the algorithm starts.

Limitation

- Use MDL (minimum description length) and binary tree Discretization.

ANT COLONY DECISION TREE(ACDT) [2]

In ACDT each ant chooses the appropriate attribute for splitting in each node of the constructed decision tree according to the heuristic function and pheromone values. The heuristic function is based on the Towing criterion, which helps ants divide the objects into two groups, connected with the analyzed attribute values.

In this way, the attribute, which well separate the objects is treated as the best condition for the analyzed node. The best splitting is observed when we classified the same number of objects in the left and right sub trees with the maximum homogeneity in the decision classes. Pheromone values represent the best way (connection) from the superior to the subordinate nodes.

- CART algorithm is used

Limitation:

- That is Only create binary tree

2.3 ANT MINER [3]

Real ant colony is that of a parallel search over several constructive computational threads based on local problem data and on a dynamic memory structure containing information on the quality of previously obtained result. Natural ants find the shortest path between a food source and the nest without using visual information.
In order to exchange information about which path should be followed, ants communicate with each other by means of a chemical substance called pheromone. As ants move, a certain amount of pheromone is dropped on the ground creating a pheromone trail. The more ants follow a given trail, the more attractive that trail becomes to be followed by other ants.

Ant-Miner Algorithm provide step by solution for discover rule .The goal of Ant-Miner is to extract classification rules from data in the form of:

IF <term1 AND term2 AND … > THEN < class >.

Algorithm consists of several steps:

Step 1:- Rule construction

Two factors:

Heuristic function (problem dependent) η

Pheromone associated with term τ.

Step 2:- Rule pruning

Improving simplicity of rule iteratively remove one-term-at-a-time Test new rule against rule-quality function:

$$Q = \frac{TP}{TP + FN} * \frac{TN}{FP + TN}$$

Step 3:- pheromone update

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t) * Q, \quad \forall i,j \in R$$

Step 4:- normalization

Normalization the amount of pheromone values at iteration for each predictor attributes.

Stopping Criteria

1. Num. of rules >= Num. of ants
2. Convergence is met
   a. Last *k ants found exactly the same rule,*
      k = No_rules_converg
3. List of discovered rules is updated
4. Pheromones reset for all trails

Limitation

It cannot cope up with the continues attribute

IV. Various Method Of Pruning And Discretization

What is Tree pruning?
Applying pruning methods to a tree usually results in reducing the size of the tree (or the number of nodes) to avoid unnecessary complexity, and to avoid over-fitting of the data set when classifying new data.
Types of tree pruning [4]
Post pruning method:
Post-pruning which removes sub trees from a 'fully growth' tree. A sub trees at a given node are pruned by removing its pruned by removing its branches and replacing it with a leaf node. The leaf is labeled with the most frequent class among the sub trees being replaced.

On line pruning method:
Online pruning is totally differing from the post pruning. in that it operates on the decision tree while is being induced. This method is useful for reducing the size of the decision tree, but always imparts a penalty on accuracy.

Discretization

- divide the range of a continuous attribute into intervals
- Some classification algorithms only accept categorical attributes.
- Reduce data size by Discretization.
- Reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals.
- Interval labels can then be used to replace actual data values.

- Three types of attributes:
  - Nominal — values from an unordered set
  - Ordinal — values from an ordered set
  - Continuous — real numbers
- Typical methods: All the methods can be applied recursively
  1) Binning (Top-down split, unsupervised)
  2) Histogram analysis (Top-down split, unsupervised)
  3) Clustering analysis (unsupervised, top-down split or bottom-up merge)
  4) Entropy based Discretization (supervised, top-down splitting)
  5) Correlation (e.g., χ^2) analysis (unsupervised, bottom-up merge)

Methods / Evaluation Criteria	Equal Width	Equal Frequency	K-means Clustering based	Entropy Based	Chi Merge based
Supervised/ Unsupervised	Unsupervised	Unsupervised	Unsupervised	Supervised	Supervised
Dynamic/ Static	Static	Static	Static	Static	Static
Global/ Local	Global	Global	Local	Local	Global
Splitting/ Merging	Split	Split	Split	Split	Merge
Direct / Incremental	Direct	Direct	Direct	Incremental	Incremental
Stopping Criteria	Fixed Bin no.	Fixed Bin no.	No further reassignment of data values to the given fixed cluster no.	Threshold / Fixed no. of intervals	Threshold / Fixed no. of intervals
Sensitive to outlier	Yes	No	Yes	No	No
Same values go to different intervals	No	Yes	No	No	No
Time Complexity to discretize one attribute of n objects	O(n)	O(n)	O(ikn) i=no of iteration and k= no of intervals	O(n log(n))	O(n log(n))

[Figure 3 : comparison of Discretization method]

V. Proposed Method

Flow of work

Step 1: inducing decision tree

Step 2: pruning method

Step 1: Inducing Decision Tree Using Aco (Main Algorithm)

1. Pre-process the relational multiclass data
   - Fill up the missing values
   - Discretization all numerical attributes
2. Make nodes representing each <attribute, value> pair.
3. Assign initial pheromone value to each node.
4. Tree $_{GB}$ = NULL.
5. Repeat following steps for each iteration
   - A. Initialize Tree $_{IB}$ = NULL.
   - B. Repeat following steps for each ant
     1. Create pruned decision tree from set of nodes.(go to step -2)
     2. If Quality (Tree) > Quality (Tree $_{IB}$)
        - Tree $_{IB}$ = Tree
   - C. Update pheromone for those nodes which are in Tree $_{IB}$.
   - D .If Quality (Tree $_{IB}$) > Quality (Tree $_{GB}$)
     - Tree $_{GB}$ = Tree $_{IB}$
6. Return Tree $_{GB}$.

Step 2: For Pruning Steps (step 1.5 b (1))

1 .Select an attribute for root node using
   Heuristic Information (Information gain/Gain ratio) and
   Current pheromone value
2. Create branch nodes from root based on distinct values
3. Repeat following steps for each branch
   - A. Let S = set of samples satisfy the branch condition
   - B. If S is empty or |S| < threshold or Remaining Attributes = 0
     Add a leaf node with label of majority class from current branch
   - C. Else if all samples of S have same class label
     Add a leaf node with label of S
   - D. Else
     Repeat this procedure again for set of the samples S.
4. Return the Tree.

VI. Conclusion and Future Work

This paper, presented a method for ant colony optimization. in that proposed method, after applying Discretization and pruning method we can get the better results. after using tree pruning and Discretization we reduced the size of the tree, and also reduce the time .that results compare with the other algorithm that is ACDT and CART, but our proposed method give better results.

In that Future work, it can cope up with the continuous attribute.

References

[1] "Inducing Decision Trees with an Ant Colony Optimization Algorithm" by Fernando E. B. Otero, Alex A. Freitas, Colin G. Johnson School of Computing, University of Kent, UK

[2] "Ant Colony Decision Trees – A New Method for Constructing Decision Trees Based on Ant Colony Optimization" by Urszula Boryczka and Jan Kozak Institute of Computer Science, University of Silesia, B¸edzi´nska 39, 41–200 Sosnowiec,Poland

[3] "Comparative Analysis of Variations of Ant-Miner by Varying Input Parameters" by Sonal P. Rami and Mahesh H. Panchal

[4] "Decision Tree Analysis using Weka Machine Learning – Project ii" by Sam Drazin and Matt Montag University of Miami

[5] "An Adaptive Discretization in the ACDT Algorithm for Continuous Attributes" by U. Boryczka and J. Kozak, In Proceedings of the ICCCI, pages 475–484, 2011.
[6]"The Ant Colony Optimization meta-heuristic" by D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization, pages 11–32, 1999.
[7] "Ant Colony Optimization and Swarm Intelligence" by Dorigo, M., et al. (eds.) ANTS 2008. LNCS, vol. 5217, Springer, Heidelberg (2008).
[8] "Special issue on ant colony optimization." By Doerner, K.F., Merkle, D., St¨utzle, T.Swarm Intelligence 3(1), 1–2 (2009).
[9]"Discretization: An enabling technique Data Mining and Knowledge Discovery" by Liu, H., Hussein, F., Tan, C., Dash, M. 6 (2002) 393–423
[10]."Ant algorithms for discrete optimization *Artificial Life*" by M. Dorigo, G. Di Caro, and L.M. Gambardella 5(2):137–172, 1999.

