Rule Based Interaction Technique with Ant Colony Optimization Algorithm

E.Balraj[1], P.Deepa[2]
[1]Assistant Professor, [2]M.E Scholar
Computer Science and Engineering,
M.Kumarasamy College of Engineering, Karur, India.
[1]balraje.cse@mkce.ac.in, [2]deepamithu85@gmail.com

Abstract-**Ant Colony Optimization (ACO) algorithms have been successfully applied to discover a list of classification rules. In general, these algorithms follow a sequential covering strategy, where a single rule is discovered at each iteration of the algorithm in order to build a list of rules. The problem in this case is not coping with the problem of rule interaction, i.e., the result of a rule affects the rules that can be discovered subsequently since the search space is modified due to the removal of examples covered by previous rules. Here, a new sequential covering strategy cAnt-MinerPB for ACO classification algorithm is proposed to mitigate the problem of rule interaction, where the order of the rules is implicitly encoded as pheromone values and the search is guided by the quality of a candidate list of rules. cAnt-MinerPB algorithm, which is the extended version of the Ant-Miner algorithm that handles continuous attributes on-the-fly during the rule construction process. The experiments are conducted using 18 publicly available data sets and results shows that the predictive accuracy obtained by a new ACO classification algorithm implementing the cAnt-MinerPB, statistically significantly higher than the predictive accuracy of state-of-the-art rule induction classification algorithms.**

Keywords - datamining, Ant colony optimization, classification, rule induction, sequential covering.

I. INTRODUCTION

Classification is one of the important data mining tasks. The main objective of this is to learn a relationship between input values and a desired output. A set of examples defined by a classification problem, where each example is explained by predictor attributes and associated with a class attribute. It consists of two phases. First phase consists of given a labeled data set—a data set consisting of examples with a known class value (label) as an input, a classification model that represents the relationship between predictor and class attribute values is built. The second phase, the classification model is used to classify unknown examples— examples with unknown class value.

Most of the classification algorithms are discussed in the previous algorithms, each example is associated with only one class value or label and class values are unrelated—i.e. there are no relationships between different class values. The above said classification problems are usually referred to as flat (nonhierarchical) single-label problems. The main problem of hierarchical multi label classifications are, examples may be associated to multiple class values at the same time and the class values are organized in a hierarchical structure (e.g. a tree or a directed acyclic graph structure). According to the data mining perspective, hierarchical multi-label classification is more challenging than flat single-label classification. Most difficult task of hierarchical multi label classification is to discriminate between classes represented by nodes at the bottom of the hierarchy than classes represented by nodes at the top of the hierarchy, since the number of examples per class tends to be smaller at lower levels of the hierarchy as opposed to top levels of the hierarchy. Another problem is, class predictions must satisfy hierarchical parent-child relationships, since an example associated with a class is automatically associated with all its ancestors' classes. Final problem is multiple unrelated classes— classes which are not involved in ancestor/descendant relationship may be predicted at the same time.

There has been an increasing interest in hierarchical classification, where in general early applications are found in text classification and recently in protein function prediction. The latter is a very active A cAnt-MinerPB Using Evolutionary Algorithm research field, given the large increase in the number of uncharacterized proteins available for analysis and the importance of determining their functions in order to improve the current biological knowledge. It is important to emphasize that in this context, comprehensible classification models which can be validated by the user are preferred in order to provide useful insights about the correlation of protein features and their functions. Concerning the problem of protein function prediction, the focus of us, an example to be classified corresponds to a protein, predictor attributes correspond to different protein features and the classes correspond to different functions that a protein can perform. Since it is known that a protein can perform more than one function and function definitions are organized in a hierarchical structure (e.g. FunCat and Gene Ontology protein functional-definition schemes), the classification problem in this case is an instance of a hierarchical multi-label problem.

II. LITERATURE SURVEY

A. Ant Colony Optimization

Ant Colony Optimization algorithms simulate the behaviour of real ants using a colony of artificial ants, which cooperate in finding good solutions to optimization problems. Every artificial ant, representing a simple agent, builds candidate solutions to the problem at hand and communicates indirectly with other artificial ants by means of pheromone values. At the same time that ants perform a global search for new solutions, the search is guided to better regions of the search space based on the quality of solutions found so far. The algorithm converges to good solutions as a result of the collaborative interaction among the ants; an ant probabilistic chooses a trail to follow based on heuristic information and pheromone values, deposited by previous ants. The interactive process of building candidate solutions and updating pheromone values allows an ACO algorithm to converge

B. MuLAM Optimization

It proposed a new ACO algorithm, named MuLAM (Multi-Label Ant-Miner), for discovering multi-label classification rules. In essence, MuLAM differs from the original Ant-Miner in three aspects, as follows. Firstly, a classification rule can predict one or more class attributes, as in multi-label classification problems an example can belong to more than one class. Secondly, each iteration of MuLAM creates a set of rules instead of a single rule as in the original Ant-Miner. Thirdly, it uses a pheromone matrix for each class value and pheromone updates only occur on the matrix of the class values that are present in the consequent of a rule. In order to cope with multi-label data, MuLAM employs a criterion to decide whether one or more.

C. hAnt Miner

This Algorithm proposed an extension of the flat classification Ant-Miner algorithm tailored for hierarchical classification problems, named hAnt-Miner (Hierarchical Classification Ant-Miner), employing a hierarchical rule evaluation measure to guide pheromone updating, a heuristic information adapted for hierarchical classification, as We as an extended rule representation to allow hierarchically related classes in the consequent of a rule. However, hAnt-Miner cannot cope with hierarchical multi-label problems, where an example can be assigned to multiple classes that are not ancestor/descendant of each other.

III. EXISTING SYSTEM

A. Introduction

A new hierarchical multi-label ant colony classification algorithm, named hmAnt-Miner (Hierarchical Multi-Label Classification Ant-Miner) is developed to overcome the aforementioned limitations. Even hmAnt-Miner shares the same underlying procedure of the hAnt-Miner algorithm as we have seen, it differs from hAnt-Miner in the following aspects:

1. The consequent of a rule is evaluated 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 tocreate a rule—only the antecedent is rep- resented in the construction graph;

2. Euclidean distance is used to define the heuristic function, where each example is represented by a vector of class membership values in the Euclidean space. Instead of using entropy in hAntMiner We can use distance measure help us to identify 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. This concept is inspired from CLUS -HMC algorithm for hierarchical multi-label classification, it is based on the paradigm of decision tree induction, rather than rule induction.

3. A distance based measure can be used to evaluate the rule quality, which is a more suitable evaluation measure for hierarchical multi-label problems;

4. Rule pruning procedure is not applied to the consequent of a rule. It is (re-)calculated when its antecedent is modified during pruning, since the set of covered examples might have changed.

B. The Consequent Rule Construction

The consequent of rule is calculated in hmAnt Miner by using the following deterministic procedure.

Consequent r,i = |SIr& label i |
----------------- ……. Equ. 1
|SIr|

|SIr& label i | - the number of examples covered by rule r that belong to the i-th class of the class hierarchy(labeli)

|SIr| - covered by a rule r

C. The distance based Heuristic Information

Heuristic information in *h*Ant-Miner involves a measure of entropy, as in the original Ant- Miner. The entropy characterizes the homogeneity. The entropy characterizes the homogeneity of a collection of examples related to the class attribute values, giving a notion of (im)purity of the class values' distribution. The more examples of the same class the lower the value of entropy will be and the purest‘is the collection of examples. It should be noted that in all calculations involving entropy, the different class labels (values) are independently evaluated—i.e. no relationship between class labels is taken into account.

In the case of Ant-Miner, which is applied to flat classification problems, the use of the entropy measure does not present a limitation, since there is no relationship between class labels. On the other hand, the same cannot be said for *h*Ant-Miner, which aims at extracting hierarchical classification rules, derived from data where the class labels are organized in a hierarchical structure. To illustrate the limitation of the entropy measure when used in hierarchical problems, let us consider the following example. Given a tree-structured class hierarchy, where labels {1, 2, 3} are children of the root label and labels {2.1, 2.2} are children of the ‘2‘label and each class label has 10 examples.

Although the entropy is calculated according to Equation (4)—across all class labels, the hierarchical relationships are not taken into account. Therefore, the entropy of a hypothetical term ‘*IPR00023 = yes*‘ which is present in 10 examples of class ‘1‘ and in 10 examples of class ‘3‘ would be the same as of a hypothetical term’*IPR00023 = no*‘ which is present in 10 examples of class’2‘ and in 10 examples of class ‘2.1‘.

The drawback in this case is that it is known that class labels ‘2‘ and ‘2.1‘ are more similar than class labels ‘1‘ and ‘3‘. Hence, it would be expected/desired that the entropy measure (or information) exploit hierarchical relationships in order to better reflect the quality of each term in the case of hierarchical classification problems. Intuitively this becomes even more important when dealing with bigger (in terms of number of class labels and depth) hierarchical structures. It should be noted that several Ant-Miner variations—as discussed have used a heuristic information based on the relatively frequency of the class predicted by the rule (or the majority class) among all the examples that have a particular term, which would also present the above limitation.

*hm*Ant-Miner employs distance-based heuristic information, which directly incorporates information from the class hierarchy. More precisely, the heuristic information of a term corresponds to the variance of the set of examples covered by the term (the set of examples that satisfy the condition represented by the term). In order to calculate the variance, the class labels of each example are represented by a numeric vector of length *m* (where *m* is the number of class labels of the hierarchy without considering the root label). The *i*-th component of the class label vector of an example is equal to 0 or 1 if the correspondent class label is absent or present, respectively.

The distance between class label vectors is defined as the Weighted Euclidean distance, given by

$$distance(\mathbf{v}_1, \mathbf{v}_2) = \sqrt{\sum_{i=1}^{m} w(l_i) \cdot (\mathbf{v}_{1,i} - \mathbf{v}_{2,i})^2} \quad \text{...Equ 2}$$

where $w(li)$ is the Weight associated with the i-th class label, v1,i and v2,i are the values of the i-th component of the class label vectors v1 and v2 , respectively. Then, the variance of a set of examples is defined as the averaged squared distance between each example‘s class label vector and the set‘s mean class vector, given by

$$variance(S_T) = \frac{\sum_{k=1}^{|S_T|} distance(\mathbf{v}_k, \overline{\mathbf{v}})^2}{|S_T|} \quad \text{....Equ 3}$$

where Sr is the set of examples covered by a term T and v is the set‘s mean class label vector. Finally, the heuristic information of a term T is given by

$$\eta_T = \frac{variance_{max} - variance(S_T)}{variance_{max}}, \quad \text{......Equ 4}$$

Where $variance_{max}$ is defined as the sum of the worst and best variance values observed across all terms in order to assign values greater than zero to the worst terms, which other- wise would avoid them to be selected by an ant. Note that the heuristic value is normalized so the smaller the value of the variance of a term T the greater its heuristic value becomes. This is analogous to the use of the entropy measure in Ant- Miner and *h*Ant-Miner, where smaller values are preferred over bigger values since they correspond to a more homogeneous partition (where the great majority of examples belong to the same class).

D. Modified Rule Pruning

Proposed algorithm does not employ a second colony in order to consequent of rules construction. So the rule pruning procedure is simplified as follows. Every time rule is submitted to a removal process of its antecedent‘s last term and has its consequent re-calculated, because the set of covered examples could change after the removal of the term. This kind of removal process is repeated until the quality of the rule decreases when its last term is removed or the rule has only one term left in the antecedent.

Let us consider the $rule_{current}$ be the rule undergoing the pruning - is considered the best rule at the beginning of pruning procedure. Every iteration of the pruning procedure, a candidate rule $rule_i$ is created by removing the last term of the antecedent of the current best $rule_{best}$ and the consequent of $rule_i$is computed according to Subsection III.B. Then, the quality measure qi for $rule_i$ is computed. Let We compare the values of qi &q_{best} , If the quality measure qi is higher than the current best quality q_{best} , $rule_i$ substitutes $rule_{best}$, completing an iteration of the pruning procedure. This procedure is repeated until $rule_{best}$ has just one term left on its antecedent or a candidate rule $rule_i$ does not improve the quality over $rule_{best}$ (i.e. $q_{best} > q_i$).

IV. PROPOSED SYSTEM

A. Introduction

It would be interesting to evaluate a more sophisticated Pittsburgh-based approach of cAnt-Miner for addressing the aforementioned limitations, for both conventional and hierarchical multi-label classification problems. The flexibility of the proposed method for handling continuous attributes, where the dynamic discretisation procedure incorporated in the rule construction process can be adapted to suit different types of classification problems. It differs from hmAnt-Miner as follows.

- Firstly, the ACO search is guided by the quality of a candidate list of rules. Therefore, pheromone values are updated based on the quality of a list of rules, in contrast to the quality of a single rule.
- Secondly, the heuristic information is updated after a rule is added to a candidate list of rules. In this way, the heuristic information is used to direct the search for different rules in order to build a list of rules since pheromone values are constant within an iteration—i.e. during the creation of a candidate list of rules

In the Pittsburgh approach, each run of the evolutionary procedure discovers a complete list of rules (the best list of rules produced over all iterations). One of the main differences between IRL and Pittsburgh approaches is that in the latter a complete list of rules, which constitutes an individual, is evaluated instead of a single rule, in order to guide the discovery process.

B. General description

The core of the algorithm is the incremental construction of a classification rule of the type IF <term1 AND term2 AND . . . >THEN <class> by an ant. Each term is an attribute-value pair related by an operator. In the current experiments, "=" as the only possible relational operator. An example term is "Color = red". The attribute's name is "color" and "red" is one of its possible values.

The total number of terms for a dataset is equal to the

$$Total\_terms = \sum_{i=1}^{a} b_i \quad \text{………Equ 5}$$

Where '*a*' is the total number of attributes (excluding the class attribute) and *bi* is the number of possible values that can be taken on by an attribute *Ai*. When an attribute has been visited, it cannot be visited again by an ant, because we do not allow conditions of the type "Color = Red" OR "Color = Green". The search space is such that an ant may pick the next term from any attribute and there is no ordering in which the attributes can be visited. An ant has to stop when all the attributes have been visited. It can also stop prematurely if the addition of the latest term makes the partial rule to cover training samples which only have the chosen class label. An example search space represented as a graph is shown in Figure 1.

In Figure 5.1, there are four attributes A, B, C and D having 3, 2, 3 and 2 possible values, respectively. An ant starts from the "Start" vertex and constructs a rule by adding conditions (attribute-value pairs or terms) for the antecedent part. After a term has been selected, all the other terms from the same attribute are prohibited for the ant. In this graph, the constructed path is C1, A3, D1 and B1.

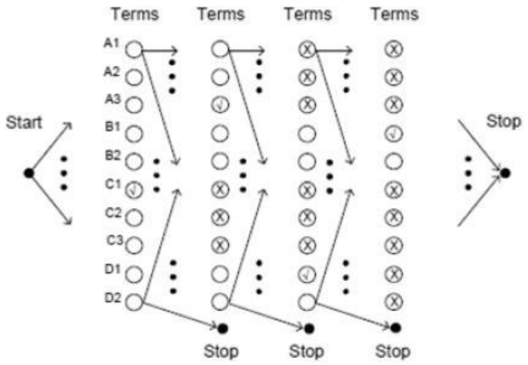

Figure 5.1 An example problem's search space represented as a graph

C. Rule Construction

The process of rule construction for a batch of ants starts by assigning them a class label. The ants construct their rules one by one, and each new ant is guided by the experience of the previous ants, available in the form of pheromone values. An ant constructs a rule by adding one term at a time. The choice of adding a term in the current partial rule is based on the pheromone values and heuristic values associated with the links from the previously added term to the candidate terms. An ant terminates its rule construction when all the instances covered by its rule are of the assigned class label or when there are no more attributes left for addition in the rule.

D. Rule Quality and Pheromone Update

When an ant has completed the construction of a rule its quality is calculated. The quality, *Q*, of a rule is computed by using confidence and coverage of the rule and is given by:

$$Q = \frac{TP}{Covered} + \frac{TP}{N} \quad \text{…… Equ 6}$$

where *TP* is the number of samples covered by the rule that have the same class label as that of the rule's consequent, Covered is the total number of samples covered by the rule, and *N* is the number of samples in the training set yet uncovered by any rule in the discovered rule set. The second portion is added to encourage the construction of rules with wider coverage. The pheromone values are updated so that the next ant can make use of this information in its search.

E. Termination of REPEAT-UNTIL loop

The REPEAT-UNTIL loop is used to construct as many rules as the user defined number of ants. After the construction of each rule its quality is determined and the pheromones on the trails are updated accordingly.

The pheromone values guide the construction of next rule. An early termination of this loop is possible if the last few ants have constructed the same rule. This implies that the pheromone values on a trail have become very high and convergence has been achieved.

Any further rule construction will most probably yield the same rule again. Hence, the loop is terminated prematurely. For this purpose, each constructed rule is compared with the last rule and a counter is incremented if both the rules are the same. If the value of this counter exceeds a threshold, then the loop is terminated.

F. Pruning of Best Rule

Rule pruning is the process of finding and removing any irrelevant terms that might have been included in the constructed rule. The rule pruning procedure starts with the full rule. It temporarily removes the first term and determines the quality of the resulting rule. It then replaces the term back and temporarily removes the second term and again calculates the quality of the resulting rule. This process continues until all the terms present in the rule are dealt with. After this assessment, if there is no term whose removal improves or maintains the quality then the original rule is retained. However, if there is one or more terms whose removal improves (or maintains) the quality of the rule then the term whose removal most improves the quality of the rule is permanently removed. In such cases, the shortened rule is again subjected to the procedure of rule pruning. The process continues until any further shortening is impossible (removal of any term present in the rule leads to decrease in its quality) or if there is only one remaining term in the rule.

G. Final Rule Set

The best rule is placed in the discovered rule set after pruning and the training samples correctly covered by the rule are removed and have no role in the discovery of other rules. A new iteration starts for discovery of the next rule if the stoppage criterion of the algorithm is not met. A final default rule is added at the bottom of the rule set. The class label for this rule is the majority class label in the set of uncovered training cases at the stoppage of the algorithm.

V. RESULT

There are two kinds of biometrics datasets has been used for this proposed algorithm.

1. Gene Ontology dataset
2. Fun cat dataset

TABLE I. INPUT DATASET [FUNCAT]

Dataset	FunCat			
	|training|	|test|	|attributes|	|classes|
cellcycle	2476	1281	77	500
Desire	2450	1275	63	500
Eisen	1857	837	79	462
Expr	2488	1291	551	500
gasch 1	2480	1284	173	500
Phenol	1009	582	69	456
Sep	2580	1339	478	500
Spo	2437	1226	80	500

TABLE II. INPUT DATASET [GENE ONTOLOGY]

Dataset	Gene ontology			
	|training|	|test|	|attributes|	|classes|
cellcycle	2473	1278	77	4126
Desire	2447	1272	63	4120
Eisen	1583	835	79	3574
Expr	2485	1288	551	4132
gasch 1	2477	1281	773	4126
Phenol	1005	581	69	3128
Sep	2568	1332	478	4134
Spo	2434	1263	80	4120

TABLE III. AVERAGE NUMBER OF CLASS LABELS IN THE HIERARCHY AND THE AVERAGE CLASS LABELS PER EXAMPLE

	Fun cat	Gene Ontology
Average number of class labels	489	3932
Average labels per Example	8.5	34.2

TABLE IV. USER DEFINED PARAMETERS USED BY OUR DATASET

Parameter	Description	Value
max uncovered examples	maximum number of uncovered examples	10
max number Iterations	maximum number of iterations	1500
rule convergence	number of iterations used to test the rule convergence	10
min examples per Rule	minimum number of covered examples per Rule	10
colony size	number of ants per iteration	30

VI. CONCLUSION

The proposed paper presents a novel ant colony algorithm; named cAnt-Miner$_{PB}$ (Pittsburgh approached cAnt-Miner$_{PB}$). Extending on the ideas of our previous hierarchical classification *hm*Ant-Miner, cAnt-Miner$_{PB}$ discovers a single global classification model, in the form of an ordered list of *IF-THEN* classification rules, which can predict all class labels from a class hierarchy at once, and examples may be assigned to multiple unrelated class labels. On account of the information from the class hierarchy, cAnt-Miner$_{PB}$ employs a distance-based measure in the dynamic discretization procedure of continuous attributes and as heuristic information in the ACO construction graph.

Our proposed work have conducted experiments comparing *cAnt-Miner$_{PB}$* against state-of-the-art decision tree induction algorithms for Hierarchical multi-label classification with most challenging sixteen bioinformatics data sets involving the prediction of protein function, with large numbers of predictor attributes and large numbers of class labels to be predicted. Class hierarchies were used in the experiments are represented in a tree (where a class label has a single parent, apart from the root label) or in a directed acyclic graph (where a class label can have multiple parents, apart from the root label) forms. We assure that *cAnt-Miner$_{PB}$* is most competitive in term of both predictive accuracy and simplicity We regard these results promising, given that *cAnt-Miner$_{PB}$*is the first ACO algorithm tailored for hierarchical multi-label classification, to the best of our knowledge.

There are several interesting directions for future research; the important one is that, the tool myra is used for implementing all types of Ant colony optimization algorithms

REFERENCES

[1] Fernando .E. B. Otero, Alex A. Freitas, and Colin G. Johnson, "A New Sequential Covering Strategy for Inducing Classification Rules With Ant Colony Algorithms". In IEEE Transactions OnEvalutionaryComputation,Vol.17,No.1, February 2013.

[2] Abdul Rauf Baig, "Correlation as a Heuristic for Accurate and Comprehensible Ant Colony Optimization Based Classifiers". IN IEEE Transactions On Evalutionary Computation, Vol.17,No.5,October 2013.

[3] Alves . R, Delgado. M, Freitas. "A Multi-label hierarchical classification of protein functions with artificial immune systems". In: Advances in Bioinformatics and Computational Biology (Proc. BSB-2008).

[4] Abdul RaufBaig, Waseem Shahzad, Salabat Khan and FarihaAltaf, "ACO based discovery of comprehensible and accurate rules from medical datasets", pp 6147-6159(2010).

[5] Blockeel .H, Schietgat .L, Struyf .J, Dzˇeroski. S, Clare .A, "Decision Trees for Hierarchical Multilabel Classification: A Case Study in Functional Genomics".In:PKDD-2006, LNAI 4213, pp 18–29.(2006)

[6] Blockeel.H,Bruynooghe .M,Dzˇeroski .S, Ramon .J, Struyf.J, "Hierarchical multi-classification". In: Dzˇeroski S, Raedt LD, Wrobel S (eds) Proceedings of the First SIGKDD Workshop on Multi-Relational Data Mining (MRDM 2002), University of Alberta, Edmon- ton, Canada,21–35. (2002)

[7] Chan. A, Freitas. A, "A new ant colony algorithm for multi-label classification with applications in bioinformatics". In: Proc. Genetic and Evolutionary Computation Conference (GECCO-2006), pp 27–34.

[8] Cesa-Bianchi .N, Zaniboni .L, Collins .M "Incremental algorithms for hierarchical classification". Journal of Machine Learning Research pp 31–54. (2004)

[9] Otero .F, Freitas .A, Johnson. C, "Handling continuous attributes in ant colony classification algorithms". In: Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Data Mining (CIDM-2009), IEEE, pp 225– 231

[10] Otero. F,Freitas .A, Johnson .C, "A Hierarchical Classification Ant Colony Algorithm for Predicting Gene Ontology Terms". In: Proceedings of the 7th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (EvoBio 2009), LNCS 5483, Springer, pp 68–79

[11] Parpinelli .R, Lopes. H, Freitas, "A Data mining with an ant colony optimization algorithm". IEEE Transactions on Evolutionary Computation 6(4):321–332. (2002)