MADAMS
MINING AND ACQUISITION OF DATA BY ANT-MINER SAMPLES

[1]P.V.Sarath Chand, [2]Dr.A.Vinaya Bab, [3]Dr.A.Govardhanu

[1]Assoc. Professor B.E(cse),M.Tech(cs),(PhD) Indur Institute of Eng & Tech .
Siddipet, India. email id sarathchand2001@yahoo.com

[2]B.E(ece)M.E(ece) M.Tech(cse) PhD(cs) Professor of CSE and Director of SCDE
J.N.T.University,Hyderabad,India.

[3]B.E(cse)M.Tech(cs) PhD(cs), Professor and HOD in CSE Department J.N.T.University Hyderabad, India.
Email govardhan_cse @yahoo.co.in.

ABSTRACT

This survey deals with the problems of job shopping samples program (JSSP) in local search. The Ant-Miner Sample is a good approach for mining data .It is the objective to make minimum spanning. In this paper both deterministic and random searching is proposed. The comparison between computational and other methods on the standards of problem instances was presented. A probability of priority samples is also implemented for initial distribution of ANTS. It also improves quality, effectiveness and parallel paradigm of solutions which can run on many instances in different sizes. This algorithm proposes different sets of jobs and machine where each machine can handle utmost one job at a time. Every job constitutes a chain of operations and should be processed during an un-interrupted time of a given machine .It proposes to find a schedule that is : an allocation of operations in the form of samples to the machines in minimum length because it is difficult to solve to its optimality[1]. The experimental results directs that the sampling of Ant-Miner algorithm can efficiently solve the problems of mining the data in JSSP environment

Keywords: *JSSP, patterns, SAM, heuristic, samples, pruning, pheromones, Mediods.*

I. INTRODUCTION

In this paper we propose a Ant-Miner sampling algorithm to explain the different tasks of data mining. A task is a goal to assign an object, machine, records or instances to a class in a given set of classes based on attribute values which are predefined in the case .The use of sampling of Ant-Miner(SAM) algorithm proposes the discovery of classification rules in the field of data mining which has to be explored[2]. The SAM is similar to ant's distribution in a scheduled sampling manner like we are aware of algorithm of clustering in data mining task.

II. SAMPLES OF ANT-MINER (SAM)

It is a system based on natural behavior of ants co-operation, co-ordination and adoptions in a classified manner .In this kind of system meta functions and heuristic functions were proposed to solve optimization problems. These functions are robust, effective and versatile in nature that is it can apply successfully to a wide range of different combinations to get optimistic solutions

III. SAM FOR DATA MINING

In this section we discuss in detail about the SAM algorithm for classification of rules called as Sampled Ant-Miner. The section is divided into six subsections namely description of Ant-Miner samples, Information measures, heuristic functions, rules of data pruning for samples, updating pheromones similar to ants and use of classifying rules[3].

ALGORITHM I: **Description of Ant-Miner samples by using pheromones.**

```
TrainingSampleSet = {all possible sample cases };/* attribute list */
DiscoveredRuleSet = [ ] /* It is initialized to empty list */
[1] Create a TrainingSampleSet T;
[2] If samples are of the same class, C then
[3]       Return T labeled with the class C.
[4] If attribute-list is empty then
[5] Return T labeled with the most common class in samples; // majority voting
[6] Select test-attribute the attribute among the attribute-list with the higher information gain;
[7] Label T with the test attributes
[8] For each known value ai of test-attribute
// partition of samples
[9] Grow from T for the condition
      Test-attribute = ai
[10] let si be the set of samples in samples for test-attribute=ai// a partition
[11] If si is empty then attach
[12] With the most common class of samples
[13] else attach attribute-list or test-attribute
```

The ant starts with an empty sample and increments by constructing classification rules by adding one slot at a time to the current sample[4]. Update the pheromones of all trails and versions by incrementing the pheromones in the trail of Ant which is proportional to the quality of the samples and decreases in other trail that is simulating pheromones evaporation. Choose the best sample among all the samples constructed by all the ants. It gives the classification of

TraingSampleSet = TrainingSampleSet – {set of all cases correctly covered samples}.

Secondly the pheromones samples constructed by Ant are pruned to remove an irrelevant data.

- The algorithm starts from training samples set. (step 1)
- If the pheromones samples are all of the same class, then the training sample set itself becomes the searching class (step 2 and 3)
- The algorithm uses an entropy based structure known as information gain as a heuristic for selecting the attribute that will be the best separate samples into individual classes (step 6).
- The attribute becomes the test or tested pheromones sample (step 7).
- In this version of algorithm all attributes are categorical, that is, discrete-valued. Continuous – valued attributes must be discredited.
- A pheromones sample is created for each known value of the test attribute and the samples are partitioned accordingly (step 8 to 10).
- The algorithm uses the same process recursively to find the samples in each partition. Once a pheromones sample has occurred, it need not be considered in any of its descendents(step 13).
- The recursive partitions stops only when one of the conditions is true

1) All pheromones samples for a given attribute belong to the same class

2) There are no remaining attributes on which the pheromones samples may be further partitioned (step 4). In this case majority voting is employed (step5).This involves converting the given pheromones sample and labeling it with the class in majority among the samples. Alternatively the class distribution of the samples may be stored.

3) There are no further pheromones samples for the test-attribute = ai (step11).In this case, a pheromones sample is created with the majority class of samples.

Information gain measures:

It is used to select the test attribute for the pheromones samples. These measures are referred to as an attribute selection measure or a measure of goodness of split. The attribute with the highest information gain or greater entropy reduction is chosen as the test attribute for the current sample[5]. This attribute minimizes the information needed to classify the samples in the resulting partitions and reflects the least randomness or impurity in these partitions. Such information –theoretical approaches minimizes the expected number of tests needed to classify an object and guarantees that a simple but not necessarily the simplest sample is found.

Heuristic functions:

The basic idea of sampling approach is to pick a random samples S of the given data D, and then search for frequent Item sets in S instead of D. In this way, we trade off some degree of accuracy against efficiency. The samples size of S is searched that the search for frequent item sets in S can be done and so only one scan of the transaction in S is required overall because we are searching for frequent item sets in S rather than in D. It is possible that we will miss some of the global frequent item sets. To lessen this possibility we use a lower support threshold that

minimum support to find the frequent item sets local to S.
It is denoted as L^S
The rest of the samples is then used to compute the actual frequencies of each item set in $L^{S.}$ A mechanism is used to determine weather all of the global frequent item sets are included in L^S
If L^S actually contains all of the frequent item sets in D , then only one scan of D is required otherwise a second pass can be done in order top find the frequent item sets that were missed in the first pass the sampling approach is especially beneficial when efficiency is utmost important such as in computationally intensive applications that must be run on a very frequent basis

Data pruning for samples:

When a training sample set is built many of the samples reflect anomalies in the training data due to noise or outliers. The Data pruning methods address the problem of over fitting the data such methods typically used statistical measures to remove the least reliable dat, generally resulting in faster classification and improvement in the ability of the samples to correctly classify independent test data[6].
There are two common approaches in data pruning

a) Pre-pruning approach – In this approach a sample data is pruned by halting its descendants early that is by deciding not to further split the partition or subsets of training sample data. Upon halting the data belongs to the class of training samples of data. It holds the most frequent class among the subset samples or the probability distribution of the samples. In constructing the sampling data the measure such as statistical significance, D^2 , information gain and so on, can be used to assess the goodness of the split. If partitioning the samples at a training set would result in a split the falls below a pre-specified threshold, then further partitioning of the given subset is halted. There are difficulties however in choosing the appropriate threshold. High thresholds could result in over simplified trees, while low thresholds could result in very little simplification.

b) Post-pruning approach – this approach removes the data samples from a fully grown training sample set. A data is pruned by removing its samples which is similar to cost complexity pruning algorithm. The lowest un-pruned data becomes a sample and is labeled by the most frequent class among its ascendant samples. The algorithm calculates the expected error rate that would occurred at the descendant sample was pruned. The next expected error rate was occurring if the sample was not pruned is calculating using the error rates for each sample, combining the weighting according to the proportion of observations along each sample. If pruning the sample leads to greater expected error rate, then the descendant is kept. Otherwise it is pruned. After generating a set of progressively pruned samples of data an independent test set is used to estimate the accuracy of each tree, the training sample set minimizes the expected error rate is preferred.

The knowledge represented in training samples set can be extracted and represented in the form of classifications IF-THEN rule. One rule is created from each path from the training sample set to the descendant's samples. Each attribute value pair along a given path forms of conjunction in the rule antecedent ("IF" part). The descendant sample holds the class prediction, forming the rule consequent ("THEN") part. The IF-THEN rules may be easier for the humans to understand, particularly if the given sample is very large enough.

Updating pheromones:

Recall that each sample of data D constitutes a segment in some path either to its descendants or ascendants followed by an ant.. At each level of journey along the path the pheromones are initialized by an ant so that when the second ant starts its searching mechanism all the paths in the respective journey contains equal amount of pheromones. The initial amount placed in the initial journey by the first ant is inversely proportional to the values of all the attributes [7]. When ever an ant constructs its rule and it should be pruned.

Algorithm is

The amount of pheromones in all segments of all paths should be updated. The pheromones updating is supported by an idea of checking similar characters

The basic idea of Ant-Miner sampling algorithm is very simple. In the case of detecting a mismatch at character T_i, P is shifted to the right to align T_i with the fifth encounter character equal to T_i, if such character exists [5].

For example, for T= vvvvzwyvvwvywyv and P=yvwvxwy, first, characters T_6=y and P_6=y, then characterT_5=w and P_5=w are compared, and then the first mismatch is found at T_4=z and P_4=x. But there is no occurrence of z in P. This means that there is no character in P to be aligned with z in T_0, i.e no character successfully matched with z. therefore; P can be shifted to the right past the mismatched characters:

```
 vv vv zwy  vv  wvy  wyv
1.  yv wv xwy
2.           yvwvxwy
```

In this way, the first four character of the text are excluded from later comparisons. Now, matching starts from the of P and position 11=4+7=(the position of mismatched character T_4)+|P|. A mismatched is found at T_{10}=v and P_5=w, and then the mismatched v is aligned with the first v to the left of mismatched P_5:

```
      vvvvzwyvvwvywyv
2   yvwvxwy.
3.           yvwvxwy
```

That is, the position in T from which the matching process starts in the third line is 13=10+3=(the position of mismatched character T_{10}=v) + (|p|-position of the right most v in P). After matching characters. T_{13} with p6 and T_{12} with P_5, v mismatch is found at T_{11}=y and P_4=x.

If we aligned the mismatch zy, y is text with the right most y in p, p would be moved backwards. Therefore, if there is a character in p equal to the mismatched character in T to the left of the mismatched character in p, the pattern p is shifted to the right by one position only:

```
  vvvvzwyvvwvywyv
3   yvwvxwy
4          yvwvxwy
```

To sum up, the three rules can be termed character occurrence rules.

1. **No occurrence rule**:
   If the mismatched character T_i appears nowhere in P, align p0 with T_i+1.
2. **Right side occurrence rule**:
   If there is a mismatch at T_i and P_j, and if there is an occurrence of character ch equal to T_i to the right of $P_{j,}$ shift P by one position.
3. **Left side occurrence rule:**
   If there is an occurrence ch equal to T_i only to the left of P_j , align T_i with P_k =ch closest to P_j .

each character in the alphabet, by how much to increment I after a mismatch is deleted. The table indexed with character and is defined as follows:

Delta1 [ch] = { |p| if ch is not is p.
Min{|p|-i-1:P_i =ch}

Otherwise

The algorithm is as follows:

```
Ant Miner pheromones (pattern P, text T)
    {
       Initialize all cells of delta1 to !p!;
       For j=0 to !p!-1
         delta1[$P_j$] = 1p!-j-1;
        i=1p!-1;
        while i<!T!
         {
           J=!P!-1;
           While j>=0 and $P_j$ == $T_i$
             {
               j--;
               j--;
              }
           If j == -1
              Return match at i+1;
             i=i+max(delta1[$T_i$], !P!-j);
         }
      Return no match;
    }
```

Classifying rules:

The classification of IF-THEN rules can be used for tracing the path from the training sample set to each and every another similar sample of data. The algorithm uses training sample set to estimate the accuracy of each rule. Since it results to the optimistic estimate of accuracy. It also employs the pessimistic estimate to compensate for the bias. Alternatively , a set of test samples independent from the training sample set can be used to estimate the rule of accuracy. A rule can be pruned by removing any conditions from its antecedents that does not improve the estimated accuracy of the rule. For, each class rules with in a class may then be ranked according to their estimated accuracy. Since it is possible that a given test sample will not satisfy any rule antecedent, a default rule assigning the majority class is typically added to the resulting rule set[8].

IV. DISCUSSION AND COMPUTATIONAL RESULTS.

Comparing Samples of Ant-Miner (SAM) with K-Medoids method

We have evaluated the performance measure of SAM with K-Medoids method, which is well known classification rule discovery algorithm. The Mediod is used for partitioning method based on the principal of minimizing the sum of the dissimilarities between each object and its corresponding reference point. The basic strategy of K-Mediods clustering algorithm is to find k clusters in 'n' objects by first arbitrarily finding a representative object (the Mediod) for each cluster. Each remaining object is clustered with the mediod to which it is the most similar[9].

Note that SAM also works similar to the K-mediods. In addition both SAM and K-Mediods construct a rule by starting with a training sample set and increments by adding one term at a time to the rule. However the construction of rules in both the algorithms are dissimilar because in K-Mediods the algorithm proceeds with iteration of replacing one of the Mediods by one of the Mediods as long as the quality fo the resulting cluster is improved. This quality is estimated using a cost function that measures the average dissimilarity between an object and the Mediod of its cluster. In SAM the object can be discovered by using the updating of pheromones samples which is a major difference with K-Mediods. The pheromones mechanism acts similar to feedback for constructing other rules. This feed back is the major characteristics in SAM and can be considered for stochastic search.

V. CONCLUSION AND FUTURE WORK

This work has been proposed for rule discovery called Samples of Ant Miner (SAM).In this paper booting of simulated Ant-Miner algorithm is presented. This algorithm is used for generating classification rules and heuristic functions. The proposed algorithm is an improvement of Ant-Miner which is used to discover the rules in classification problems. The main theme of this algorithm is to discover the classification rules in data sets. The algorithm is based both on the research and to the real ant colonies and on mining the data concepts and their principles. We have compared the performances of Sampling of Ant-Miner and well known K-Mediods algorithm in six public domain data set samples. The result showed that concerning predictive accuracy and Sampled Ant-Miner obtained some better results in four data set sample, where as K-Mediods obtained a considerably better result in one data set sample. In the remaining data samples both algorithms obtained the same accuracy which was predicted. There fore one can say that Ant-Miner Samples or Sampling Ant-Miner is roughly equal to K-Mediods with respective to predictive accuracy and the other hand Ant-Miner sampling has consistently found very much simpler and smaller rule lists than K-Mediods. There fore, this algorithm seems particularly advantageous when it is important to minimize the number of discovered rules and rule terms and conditions. In order to improve the comprehensibility of the knowledge. It can be stated that it is very important in many data mining application, where the discovery of data or knowledge is visible to the human user as a support for intelligent decision making as discussed earlier. To impart the direction for future research are

(1) It would be interesting to extend as Ant-Miner sampling algorithm with continuous attribute rather than the acquiring the attribute in discredited in a preprocessing steps.

(2) It would be interesting to investigate the accuracy of other kinds of heuristic functions and pheromone updating.

REFERENCES:

[1] A. Silberschatz and A.Tuzhilin what makes pattern interesting in knowledge discovery systems. IEEE Trans data and Knowledge Engineering, 8, 970-974 Dec 1996.

[2] M.Stone cross-validatory choice and assessment of statistical predictions. General of the royal statistical society.

[3] W. Siedlecki and J.Sklansky on automatic feature selection.Int.J. Of Pattern recognition and Artificial Intelligence 2. 197-220, 1988.

[4] E. Bonabeau, M. Dorigo and G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems. New York, NY: Oxford University Press, 1999.

[5] J. Catlett, “Overpruning large decision trees,” In: Proceedings International Joint Conference on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann, 1991.

[6] Adam Drozdek “ Simplifying large decision trees,” In: Proceedings International Joint Conference on Data mining. San Francisco,

[7] M.Bohanec and I. Bratko, “Trading accuracy for simplicity in decision trees,” Machine Learning, vol. 15, pp. 223-250, 1994.

[8] L.A. Brewlow and D. W. Aha, “Simplifying decision trees: a survey,” The Knowledge Engineering Review, vol. 12, no. 1, pp. 1-40, 1997.

[9] M.P.Oakes,”Ant-Colony Optimization for stylometry: The federalist paper. Internation conference on soft computing Nov 2004