International journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.1., Issue.2., 2013

RESEARCH ARTICLE

ISSN: 2321-7758

A SURVEY ON DIFFERENT ALGORITHMS OF ASSOCIATION RULE MINING AND ANT COLONY OPTIMIZATION

NN DAS, ANJALI SAINI*

* *Computer Science and Engineering, Itm University, Gurgaon, India*

Article Received: 14/11/2013 | Article Revised on: 22/11/2013 | Article Accepted on:28/11/2013

ANJALI SAINI *

NN Das

ABSTRACT

Association Rule Mining is one of the most popular techniques used in data mining. This paper aims at giving an overview to some of the previous researches done in this topic, evaluating the current status of the field, and envisioning possible future trends in this area. The theories behind association rules are presented at the beginning. We also present an efficient mining based optimization techniques for rule generation i.e. ACO(Ant Colony Optimization) for optimizing the association rules.

Keywords - Association Rule Mining, Apriori , AprioriTid , AprioriHybrid , ACO

INTRODUCTION

Discovering Association Rules is at the heart of data mining. Association Rule Mining(ARM) is one of the most used research in data mining. It is used for discovering hidden valuable relationship or useful information between items. It can also be studied in terms of market basket analysis which is like analysis of customer of purchasing behavior. Association rules can also be used in various areas such as retail industry, biological and financial data analysis, telecommunication industry etc.

The problem of discovering association rules can be generalized into two steps:

(1) Finding large itemsets.

(2) Generating rules from these itemsets.

Data Mining field comprises of four main disciplines: *Statistics*: defines tools for measuring significance in the data, *Machine learning*: provide algorithm to induce knowledge from the data, *Artificial intelligence*: involve knowledge

for encoding and search techniques, *Data management and databases:* provides an efficient way of accessing and maintaining data.Swarm Intelligence(SI) is an artificial intelligence technique which is a collective behavior of trustworthy,decentralized,self-organized system,natural or artificial.One of the SI techniques is ACO(Ant Colony Optimization).ACO is an meta heuristic algorithm inspired in the cooperative foraging behavior of ants to find and exploit the food source that is nearest to nest.ACO can be applicable to the solution of combinatorial optimization problem.

ASSOCIATION RULES

The following is a formal statement of the problem[3]:

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called items. Let D be set of transactions, where each transaction is set of items . Associated with each transaction is a unique identifier, called its TID. We say that a transaction T contains X, a set of some items in I, if X C T.

An association rule is an implication of the form-

X → Y,where

$$X \subseteq I, Y \subseteq I \text{ and } X \bigcap Y = \phi$$

- Association rule X→Y has confidence c if c% of transactions in D that contain X also contain Y.
- Association rule X→Y has support s,if s% of transactions in D contain X U Y.

The problem of mining association is to generate all association rules that have support and confidence greater than the user-specified minimum support and minimum confidence in a given set of transactions D.

ALGORITHMS: The different algorithms used in data mining are given below-

1. APRIORI ALGORITHM: Apriori is a great improvement in the history of association rule mining, Apriori algorithm was first proposed by Agrawal in [Agrawal and Srikant 1994]. Apriori is more efficient during the candidate generation process for two reasons, Apriori employs a different candidates generation method and a new pruning technique.

It uses prior knowledge of frequent itemset properties.It is an iterative algorithm known as level-wise search.The search proceeds level-by-level as follows:

- First determine the set of frequent 1-itemset; L1
- Second determine the set of frequent 2-itemset using L1: L2
- Etc.

PSEUDO-CODE

C_k: Candidate item set of size k

L_k : frequent item set of size k

L_1 = {frequent items};

for (k = 1; L_k !=Æ; k++) **do**

-C_{k+1} = candidates generated from L_k;

- **for each** transaction t in database **do**

Increment the count of all candidates in C_{k+1} that are contained in t;

endfor;

- L_{k+1} = candidates in C_{k+1} with min_support

endfor;

return $\cup_k L_k$;

Challenges:

- It includes multiple scans of transaction database
- number of candidates
- Tedious workload of support counting for candidates.

2. APRIORITID ALGORITHM:

The interesting feature of this algorithm is that it uses the database only once.[not used for counting support after first pass].The set $C^{\wedge}_k$ is used for that purpose.Each member of set is in form < TID, $\{X_k\}$ >,where X_k are potentially large k-items in transaction TID.For k=1, $C^{\wedge}_1$ is the database.

$L_1 = \{large\ 1\text{-}itemsets\}$
$C^{\wedge}_1 = database\ D;$
For $(k = 2; L_{k-1} \neq \phi; k++)$ do begin
  C_k = apriori-gen(L_{k-1});
  $C^{\wedge}_k = \phi;$
  forall *entries* $t \in C^{\wedge}_{k-1}$ do begin
    $C_t = \{c \in C_k | (c - c[k] \in t.set-of-items$
      $\wedge (c - c[k-1]) \in t.set-of-items\};$
    forall candidates $c \in C_t$ do
      $c.count++;$
      if $(C_t \neq \varphi)$ then $C^{\wedge}_k += < t.TID, C_t >;$
    end
  end
  $L_k = \{c \in C_k | c.count \geq minsup\}$
end
$Answer = \bigcup_k L_k;$

Advantages:

- $C^{\wedge}_k$ could be smaller than the database. If a transaction does not contain k-itemset candidates, than it will be excluded from $C^{\wedge}_k$
- For large k, each entry may be smaller than the transaction. The transaction might contain only few candidates

Disadvantage:

For small k, each entry may be larger than the corresponding transaction because an entry includes all k-itemsets contained in the transaction.

3. ALGORITHM APRIORI HYBRID:

Based on the observations, we can design a hybrid algorithm, which we call AprioriHybrid, that uses Apriori in the initial passes and switches to AprioriTid when it expects that the set c^k at end of the pass will fit in memory. At end of the current pass,we have the counts of the candidates in c^k. From this, we estimate what the size of c^k would have been if it had been generated. This Size, in words, is (∑candidates c E ck Support(C) + number of transactions). If c^k in this pass was small enough to fit in memory, and there were fewer large candidates in the current pass than the

previous pass, we switch to AprioriTid. The latter condition is added to avoid switching when c^k in the current pass fits in memory but c^k in the next pass may not.

We have run the performances tests with the datasets and AprioriHybrid performs better than Apriori ana AprioriTid in almost all the cases.AprioriHybrid did up to 30% better than Apriori,and up to 60% better than AprioriTid.

ANT COLONY OPTIMIZATION:

Ant Colony Optimization (ACO), which was introduced in the early 1990s as a novel technique to solve hard combinatorial optimization problems[9]. ACO is an optimization algorithm inspired in the collective foraging behavior of ants to find and exploit the food source that is nearest to the nest. Stigmergy, a type of indirect communication mediated by modification of the environment. Communications among individuals or between individuals and the surroundings is based on the use of chemicals formed by the ants called pheromone [10].

The [11] design of an ACO algorithm implies the specification of the following aspects:

- An appropriate manifestation of the problem, which allows the ants of incrementally build solutions through the use of a probabilistic transition rule, based on the amount of pheromone in the trail and on a local problem dependent heuristic
- A method to enforce the construction of valid solutions, i.e., solutions that are legal permissible in the real world situation corresponding to the problem definition:
- A problem dependent heuristic function(η) that measures the significance of items that can be added to the current partial solution;
- A rule for pheromone updating, which specifies how to fine-tune the pheromone trail (τ).
- A probabilistic transition rule based on the value of the heuristic function (η) and on the contents of the pheromone trail (τ) that is used to iteratively construct a solution.

ACO Algorithm

1. Each ant searches for a minimum cost feasible partial solution.
2. An ant k has a memory Mk that it can use to store information on the path it followed so far. The stored information can be used to build feasible solutions, evaluate solutions and retrace the path backward.
3. An ant k can be assigned a start state sks and more than one termination conditions ek.
4. Ants start from a start state and move to feasible neighbor states, building the solution in an incremental way. The procedure stops when at least one termination condition ek for ant k is satisfied.
5. An ant k located in node i can move to node j chosen in a feasible neighborhood Nki through probabilistic decision rules. This can be formulated as follows:

An ant k in state sr =< sr-1; i > can move to any node j in its feasible neighborhood Nki , defined as Nki = {j | (j Є Ni) Λ (< sr, j >Є S)} sr Є S, with S is a set of all states.

6. A probabilistic rule is a function of the following.

a) The values stored in a node local data structure Ai = [aij] called ant routing table obtained from pheromone trails and heuristic values,

b) The ant's own memory from previous iteration, and

c) The problem constraints.

7. When moving from node i to neighbor node j, the ant can update the pheromone trails τij on the edge (i, j).
8. Once it has built a solution, an ant can retrace the same path backward, update the pheromone trails and die.

CONCLUSION

In this paper, we surveyed the list of existing association rule mining techniques. The topic of discovering association rules has been studied over couple of decades. Most of the foundation researches have been done. A lot of attention was focus on the performance and scalability of the algorithms. In the coming decades, the trend will be to

turn the attention to the application of these researches in various areas of our lives, e.g. genetic research, medicine, homeland security etc. Therefore, we motivated this field of research, gave a more formal definition of the terms used herein and presented a brief overview of currently available association rule mining methods.

REFERENCES

[1]. Jiawei Han, Jian Pei, Yiwen Yin: Mining Frequent Patterns without Candidate Generation, ACM SIGMOD, 2000 .
[2]. Rakesh Agrawal, Ramakrishnan Srikant: Fast Algorithms for Mining Association Rules (1994).
[3]. Agrawal, R., Imielinski, T., and Swami, A. N. 1993. Mining association rules between sets of items in large databases. In Proceeding of1993 ACM SIGMOD International Conference on Management of Data, P. Buneman and S. Jajodia, Eds. Washington, D.C., 207-216.
[4]. Han, J. and Pei, J. 2000. Mining frequent patterns by pattern-growth: methodology and implications. ACM SIGKDD Explorations Newsletter 2, 2, 14-20.
[5]. Parepinelli, R.S., Lopes, H.S., & Freitas, A., —Data Mining with an Ant Colony Optimization algorithm‖ , IEEE transactions on evolutionary computation, vol. 6, no. 4, August 2002
[6]. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. Research Report RJ 9839, IBM Almaden Research Center, San Jose, California, June 1994.
[7]. S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels. Self-organized Shorcuts in the Argentine Ant. Naturwissenschaften, 76:579–581, 1989.
[8]. M. Dorigo, Gianni Di Caro, and Luca M. Gambardella. Ant Algorithms for Discrete Optimization. Technical Report Tech. Rep. IRIDIA/98-10, IRIDIA, Universite Libre de Bruxelles, Brussels, Belgium, 1998.
[9]. Marco Dorigo and Thomas Stutzle, —Ant Colony Optimization‖ , Edition 2004 and ISBN-81-203-2684-9.
[10]. Bonabeau, E.,Dorigo,M., & Theraulaz, G.(1999),‖ Swarm Intelligence: from Natural to Artificial Systems‖ ,New York;Oxford University Press
[11]. M. Dorigo and M. Maniezzo and A. Colorni. The Ant Systems: An Autocatalytic Optimizing Process. Revised 91-016, Dept. of Electronica, Milan Polytechnic, 1991.
[12]. M. Dorigo and G. Di Caro. New Ideas in Optimisation. McGraw Hill, London, UK, 1999.