Rule Mining in Medical Domain Using Swarm based Techniques

Varinder Singh[1]

Assistant professor
[1]BAM Khasla College, Dept. of Computer Science,
Garhshankar, Punjab, India
varindersaini1990@gmail.com

Abstract: In this paper, swarm intelligence based technique for mining rules over a medical database have been used. Rules are a suitable method for representing real world medical knowledge because of their simplicity, uniformity, transparency, and ease of inference. Swarm Intelligence (SI) has been applied to the rule mining process as its dynamic nature provides flexibility and robustness to process of rule mining. Traditional methods of rule mining generate a large number of rules with too many terms, making the system unusable over medical data. In this paper, the attempt is to use SI as a novel method for discovering interesting rules in the medical domain. The performance of three different swarm based techniques has been compared by observing the output rules of rule sets used to classify data.

Section 1 introduces the concept of swarm intelligence and rule mining and how these can be combined. Issues that arise in mining medical data are also briefly listed. Section 2 describes conventional rule mining techniques and states the motivation behind using swarm intelligence for rule mining and classification. Section 3 describes the various SI based algorithms that have been implemented in our study. Section 4 describes the details of the experiment. Section 5 presents the results of the practical experiment followed by conclusions and future scope in section 6.

Keywords: Swarm intelligence, Data Mining, Soft Computing, Medical domain, Association rule mining, Ant colony optimization, Particle swarm optimization, rule quality.

1.1.Introduction

Swarm Intelligence is an innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates. Data Mining is an analytical process designed to explore large amounts of data for consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data.

Association rules were proposed for expressing knowledge in a symbolic way. Association rule mining aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in data repositories. Rules have advantages of simplicity, uniformity, transparency, and ease of inference, that have made them one of the most widely adopted approaches for representing real world medical knowledge.

Other structures like Decision trees and Bayesian networks are shown to be not as adequate for medical systems as association rules. Association rules generally include simpler predictive rules, they work well with user-binned attributes, rule reliability is higher and rules generally refer to larger sets of patients [2].

Association rules have been used in medical domain to aid in infection detection and monitoring, to understand what drugs are co-prescribed with antacids, to discover frequent patterns in gene data, to understand interaction between proteins, to find cooccuring diseases, for pharmacovigilance [1], diagnosing hyperlipidemia [3], to determine candidates for temporal lobe surgery [4], and to detect common risk factors in pediatric diseases.

The main issue in mining association rules on a medical data set is the large number of rules that are discovered, most of which are irrelevant. Such a large number of rules make search slow and interpretation by the domain expert difficult. The commonly used interestingness measures of support and confidence cannot be effective in pruning the resultant rule set because an association that holds true for even a small number of patients, can be significant and should be considered. Also finding rules with a large number of terms or conditions on attribute values is not uncommon [5]. Some other issues in medical data [6] [7] include distributed and uncoordinated data collection, strong privacy concerns, diverse data types (image, numeric, categorical, missing information), complex hierarchies behind attributes and a comprehensive knowledge base.

Swarm intelligence can be an effective tool in mining rules of high quality. The dynamic essence of SI provides flexibility and robustness to process of rule mining. With

full control on the extracted rules, SI is a suitable approach to satisfy medical systems requirements [8]. Issues related to data collection and privacy can be handled with suitable additions to the core mining algorithm based on swarm techniques.

2.1.Conventional Rule Mining

An association rule can be defined as:

Let I a set of m distinct attributes, T be a transaction that contains a set of items such that T is a subset of I, D be a database with different transaction records Ts. An association rule is an implication in the form of X ->Y, where X, Y are subsets of I and $X \cap Y = \{null\}$. Any set of items is called an itemset. X is called antecedent while Y is called consequent, the rule means X implies Y with a certain degree of support and confidence. If the consequent is a 1-itemset, it can function as a class label and the rule can be used for classification purpose.

Support(s) of an association rule is defined as the percentage/fraction of records that contain X U Y to the total number of records in the database. Confidence of an association rule is defined as the percentage/fraction of the number of transactions that contain X U Y to the total number of records that contain X. Confidence is a measure of how strong the association rule is.

Rule mining problem is usually decomposed into two sub problems. One is to find those itemsets whose occurrences exceed a predefined threshold in the database (defined by the support); those itemsets are called frequent or large itemsets. The second problem is to generate association rules from those large itemsets with the constraints of minimum confidence. Generally, an association rule mining algorithm contains the following steps:

- The set of candidate k-itemsets is generated by adding one item at a time to large (k-1) itemsets generated in the previous iteration.
- Supports for the candidate k-itemsets are generated by a scan over the database.
- Itemsets that do not have the minimum support are discarded and the remaining itemsets are called large k-itemsets.

This process is repeated until no more large itemsets are found. Most approaches to rule mining have been based on candidate generation using an Apriori [9] style algorithm or FP-tree [10] style approaches to mine rules without candidate generation. Efforts have been made to improve the performance of these techniques by either i) reducing the number of passes over the database [11] [12], or ii) sampling data [13] [14] [15], or iii) adding extra constraints on the structure of rules [16] [17] or iv) parallelization of operations [18] [19] [20] or v) a combination of these. But these different strategies still do not return accurate results in a reasonable time.

SI based techniques perform a global search and cope better with attribute interaction than the greedy rule induction algorithms often used in data mining. The improvements are

reflected in rules output to the user and classification systems constructed using these rules.

3. Swarm intelligence based rule mining

A swarm can be viewed as a group of agents cooperating to achieve some purposeful behavior and achieve some goal. The agents use simple local rules to govern their actions and via the interactions of the entire group, the swarm achieves its objectives.

A type of self-organization emerges from the collection of actions of the group. An autonomous agent is a subsystem that interacts with its environment, which probably consists of other agents, but acts relatively independently from all other agents [21]. The autonomous agent does not follow commands from a leader, or some global plan [22].

(PSO) and Ant Colonies Optimization (ACO) are currently the most popular algorithms in the area of swarm intelligence.

3.1. Ant Colony Optimization (ACO)

Each ant can be regarded as an agent that incrementally constructs/modifies a solution for the target problem. In our case the target problem is the discovery of rules. The main steps of the ACO algorithm are given below:

TrainingSet = {all training cases};

DiscoveredRuleList = [];

WHILE (TrainingSet >=□Max_Uncovered_Cases)

i = 1; /* ant index */

No_Ants_Converg = 1;

Initialize all trails with the same amount of pheromone;

REPEAT

Ant$_i$ starts with an empty rule and incrementally constructs a classification rule R$_i$, by adding one term at a time to the current rule;

Prune rule R$_i$;

Update the pheromone of all trails, by increasing pheromone in the trail followed by Ant$_i$ and decreasing pheromone in the other trails;

IF (R$_i$ is equal to R$_i$ − 1) /* update convergence test */ THEN No_Ants_Converge = No_Ants_Converge + 1; ELSE No_Ants_Converge = 1;

END IF

i = i + 1;

UNTIL (i >= □No_of_Ants) OR

(No_Ants_Converg >= □No_Rules_Converg)

Choose the best rule R_{best} among all rules R_i constructed by all the ants;

Add rule R_{best} to DiscoveredRuleList;

TrainingSet=TrainingSet - {set of cases correctly covered by R_{best}};

END WHILE

Initially, for all attributes i and their possible values j, a given initial amount of pheromone is deposited in the respective position. This initial normalized amount is proportional to the total number of values of all attributes, and is given by Eq. (1).

$$\tau_{i,j}(t=0) = 1/\Sigma_i{}^a b_i \qquad (1)$$

Where a is the total number of attributes and b_i is the number of possible values of attribute i. At the completion of a rule, the amount of pheromone in the $i|j$ (attributes i | values j) that constitute the rule must be updated. The pheromone updating is performed as follows: for all terms $i|j$ belonging to the rule created by the ant, the amount of pheromone is increased proportionally to rule quality Q, according to Eq. (2) and (3).

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t) * Q, \text{ for all } i \mid j \in \text{ to the rule} \qquad (2)$$

Where rule quality,
Q=(TruePos/(TruePos+FalseNeg))*(TrueNeg/(FalsePos+TrueNeg))   (3)

The heuristic criterion for choosing an attribute i with value j is based on the amount of information (entropy measurement) associated with the attribute value pair. For rule pruning, one condition is removed iteratively from the rule and Q is measured. The condition whose removal leads to maximum improvement in Q is removed [23] [24]. ACO works well with nominal and categorical attributes. Continuous (real valued) attributes have to be discretized first.

3.2. Particle Swarm Optimization (PSO)

PSO algorithms make use of particles moving in an n-dimensional space to search for solutions for an n-variable function optimization problem. A particle decides where to move next, considering its own experience, which is the memory of its best past position, and the experience of its most successful neighbor. The pseudo-code of basic PSO is as follows:

Initiate_Swarm()

{

For p=1 to number of particles

Evaluate (p)

Update_past_experience (p)

Update_neighbourhood_best(p,k)

For d=1 to number of Dimensions

Move(p,d)}

Until Criterion

}

The particle's previous best position (P_{id}) and the best position in the neighborhoods (P_{gd}) are maintained and updated. There is also a velocity (v_{id}) associated with each dimension, which is an increment to be made, in each iteration, to the dimension associated (Eq. (4)), thus making the particle change its position in the search space.

$$v_{id}(t)=\chi(v_{id}(t-1)+\varphi_{1id}(P_{id}-x_{id}(t-1))+\varphi_{2id}(P_{gd} -x_{id}(t-1)))$$
$$x_{id}(t)=x_{id}(t)+ v_{id}(t)$$
$$(4)$$

φ_1 and φ_2 are random weights defined by an upper limit, χ is a constriction coefficient. The general effect of Eq. (4) is that each particle oscillates in the search space between its previous best position and the best position of its best neighbour, hopefully finding new best points during its trajectory.

The formula used to evaluate a rule and therefore set its quality is given by Eq. (5)

$Q(X)= sensitivity*specificity, \text{ if } 0.0<=x_i<=1.0 \text{ for all } i\in d$
Otherwise $Q(X)=-1.0$ (5)

This formula penalizes a particle which has moved out of legal values, assigning it with negative value and thereby, forcing it to return to the search space. PSO is low in spatial complexity [25]

3.3. Combined ACO/PSO

It uses a sequential covering approach to discover one classification rule at a time according to the following algorithm.

RS = {} □/* initially, Rule Set is empty */

FOR EACH class C

TS = {all training samples belonging to all classes}

WHILE (number of uncovered training examples of class C > MaxUncovExampPerClass)

Run the PSO/ACO algorithm to discover the best rule predicting class C, called BestRule

RS = RS U □BestRule

TS= TS−{training samples correctly covered by discovered

rule}   END WHILE

END FOR

END FOR

Each particle represents the antecedent of a candidate classification rule. The rule's class is fixed for all the particles in each run of the algorithm since each run of the algorithm aims at discovering the best rule for a fixed class.

This has the advantage of avoiding the problem of having different particles predicting different classes in the same population. Continuous values can be directly represented as a component of the vector associated with a particle and processed using the standard PSO. A simple approach would be to define upper and lower bounds for the continuous attribute in the rule. A particle contains a number of pheromone matrices equal to number of categorical attributes in the data set. Each pheromone matrix contains values for pheromones for each possible value that that attribute can take plus a flag value (the indifference flag) indicating whether or not the attribute is selected to occur in the decoded rule. Updating a particle's pheromone (the probabilities of choosing attribute values) is done as follows:

$$\tau_{cij} = \tau_{cij} + (\varphi_1 * Q_c), \text{ for all } ij \text{ belongs to } CurrentRule \quad (6)$$

$$\tau_{cij} = \tau_{cij} + (\varphi_2 * Q_P), \text{ for all } ij \text{ belongs to } BestPastRule \quad (7)$$

$$\tau_{cij} = \tau_{cij} + (\varphi_3 * Q_1), \text{ for all } ij \text{ belongs to } BestLocalRule \quad (8)$$

Where τ_{cij} is the amount of pheromone in the current particle c, for attribute i, for value j. Q is the quality of the rule as given by Eq. (3). φ is a random learning factor in the range 0..1.

The population is initialized in positions with nonzero qualities by taking a record from the class to be predicted and using its terms (attribute values) as the rule antecedent. Then a pruning procedure based on term quality is initially applied, and for other iterations a method similar to ACO's pruning is applied for the final rule produced by each run of the hybrid PSO/ACO algorithm [26].

3.4. ACO/PSO with new Quality Measure-PF
This algorithm uses a sequential covering approach similar to ACO/PSO to discover one classification rule at a time.

RS = { }

FOR EACH class C

TS = {All training examples belonging to any class}

WHILE (Number of uncovered training examples belonging to class C > *MaxUncovExampPerClass*)

Run the NR algorithm to discover best nominal rule predicting class C called *Rule*

Run the standard PSO algorithm to add continuous terms to *Rule*, and return the best discovered rule *BestRule*

Prune *BestRule*

 RS = RS ∪ *BestRule*
TS = TS −{training examples covered by discovered rule}

ENDWHILE END

FOR

Order rules in RS by descending Quality

Prune RS removing unnecessary terms and/or rules

A single iteration of this loop only discovers rules based on nominal attributes, returning the best discovered rule. For the continuous part of the rule, a conventional PSO algorithm (applied only to numeric attributes) with constriction is used. The vector to be optimized consists of two dimensions per continuous attribute, one for an upper bound and one for a lower bound. At every particle evaluation, the vector is converted to a set of terms and added to *Rule* produced by the algorithm for fitness evaluation. If two bounds cross over, both terms are omitted from decoded rule, but *Personal Best* position is still updated in those dimensions using Eq. (10)

$$v_{id} = \chi (v_{id} + c_1\varphi_1(P_{id} - x_{id}) + c_2\varphi_2(P_{gd} - x_{id})))$$

$$x_{id} = x_{id} + v_{id} \quad (10)$$

To improve the performance of the PSO algorithm, each particle's initial position is set to a uniformly distributed position between the value of a randomly chosen seed example's continuous attribute and that value added to the range for that attribute (for upper bound) and at a uniformly distributed position between an example's value and an example's value minus the range for that attribute (for lower bound). The particles are prevented from fully converging using the Min-Max system. After the *BestRule* has been generated it is then added to the rule set after being pruned using ACO's pruning method. But since this is computationally expensive, ACO pruning is applied only if the number of terms is less than a fixed number. Nominal attributes are handled by the NR algorithm as follows:

Initialise population

REPEAT for MaxInterations

FOR every particle x

Set Rule R_x = "IF {null} THEN C"

FOR every dimension d in x

Use roulette selection to choose whether the state should be set to off or on. If it is on then the corresponding attribute-value pair set in the initialization will be added to R_x; otherwise (i.e., if off is selected) nothing will be added.

LOOP

Calculate Quality Q_x of R_x

P = x's past best state

Q_p = P's quality

IF $Q_x > Q_p$

Q_p = Q_x P

= x END

IF LOOP

FOR every particle x

P = x's past best state

N = the best state ever held by a neighbour of x according to N's

quality Q_N

FOR every dimension d in x

IF $P_d = N_d$ THEN pheromone entry corresponding to the value of

N_d in the current x_d is increased by Q_p

ELSE IF P_d = off AND seeding term for $x_d \neq N_d$ THEN

pheromone entry for the off state in x_d is increased by Q_p

ELSE

pheromone entry corresponding to the value of N_d in the current

x_d is increased by Q_p

END IF

Normalize pheromone entries

LOOP

LOOP LOOP

RETURN best rule discovered

Each particle has four neighbours. Initially, pheromone state in each dimension is set to 0.9 for on and 0.1 for off. Quality, Q is defined using Precision as given by Eq. (11):

Laplace-corrected Precision = $(1+TP)/(1+TP+FP)$

If TP<MinTP, Q=Laplace-Corrected Precision* 0.1,

ELSE Q=Laplace-Corrected Precision   (11)

where MinTP is the least number of correctly covered examples that a rule has to cover [27].

4. Experimental setup

4.1 Database

The data sets used for rule mining are from the STULONG data set. STULONG is an epidemiologic study carried out in order to elaborate the risk factors of atherosclerosis in a population of middle aged men. Our study focuses on identifying the relationship between alcohol intake, smoking, cholesterol and hypertension. The first three categories of attributes can then be used to characterize a patient as hypertensive or not. Data corresponding to 1417 persons and 14 attributes has been considered. These attributes were manually extracted. Continuous attributes were discretized for ACO using field knowledge of medical experts. Nominal and categorical data was cleaned to handle missing values. ACO/PSO algorithms can handle both nominal and continuous attributes.

4.2. Setting of Variable Values

For ACO, the following parameter values were taken: Number of Ants=2000, Minimum number of records per rule=15, maximum number of uncovered records=20 and number of rules to test ant convergence=30. For PSO/ACO and

PSO/ACO with PF, number of particles=100 and number of iterations=200. For PSO/ACO with PF, ACO pruning was used if rule has less than 20 terms. The value for minimum number of true positives=15, constriction factor χ=0.729, social and personal learning coefficients, c1=c2=2.05. Maximum number of uncovered examples per class was set to 20. Also, the constant factor of 0.1 in Eq. (11) was replaced with 0.4 in order to penalize false positives more severely, as this is desirable in medical domain. These values are not optimized.

5. Results

The first criterion used to analyze the performance of the various implemented techniques is predictive accuracy, defined in terms of cross validation accuracy rate, which in turn equals quotient between number of test cases correctly classified and the total number of test cases. A 10-fold cross validation was used with value of k=10. The other two criteria for performance evaluation are the number of rules in a rule set and the number of attribute value combinations or conditions per rule.

Table 1 summarizes the results obtained by the ACO, combined ACO/PSO and ACO/PSO with Precision Fitness algorithms.

Table 1: Comparison between ACO, ACO/PSO and ACO/PSO with PF

	Accuracy	No. of rules in rule set	No. of conditions in rule
ACO	80.75%	18.20± 0.87	13.80± 1.89
ACO/PSO	78.1%	15.1± 2.27	10.84± 0.38
ACO/PSO with new Q	87.43%	12.3± 1.0	7.81± 0.16

6. Conclusions and future work

The rule quality can be viewed in terms of its accuracy and comprehensibility. A rule will be interesting to a medical practitioner if it is accurate and easily understood. All three techniques studied provide accuracy comparable to other non SI based mining approaches. A system for rule mining over medical data needs to include and consider rules with small values of support without making the system unwieldy. A system generating large number of rules or rules with too many conditions in the antecedent tends to confuse the end user and is not usable for medical knowledge discovery. ACO/PSO with new quality measure of fitness performs the best in terms of comprehensibility and accuracy. This method also penalizes false positives severely, which is a desirable property for data mining in the medical domain. One drawback of the approach is the complexity of the algorithm. One possible further research direction is to introduce new data structures to reduce execution time. Certain domain specific constraints can be applied in the preprocessing phase to reduce the input data size. Also, further work needs to be carried out to arrive at the optimal values for variables in swarm intelligence algorithms.

7. References

[1] Singh, V., & Sood, M. M. "Krill Herd Clustering Algorithm using DBSCAN Technique" IJCSET Pp 197-201.

[2] Singh, V.2013 "Service Effectiveness and user requirement for Cloud Service Allocation" IJSER Pp 332-335.

[3] Sengul Dogan and Ibrahim Turkoglu , Diagnosing Hyperlipidemia using association rules, Mathematical and Computational Applications, Vol. 13, No. 3, pp. 193-202, 2008.

[4] M. Ghannad-Rezaie, H. Soltanain-Zadeh, Senior Member, IEEE, M.-R. Siadat, K.V. Elisevich, Medical Data Mining using Particle Swarm Optimization for Temporal Lobe Epilepsy, 2006 IEEE Congress on Evolutionary Computation.

[5] Sotiris Kotsiantis, Dimitris Kanellopoulos, Association Rules Mining: A Recent Overview GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), 2006, pp. 71- 82.

[6] John F Roddick, Peter Fule, Warwick J. Graco, Exploratory Medical Knowledge Discovery Experiences and Issues, Health Insurance Commission Australia.

[7] Jean-Marc Trémeaux, Yan Liu , Mining for association rules in medical data , 2006.

[8] M. Ghannad-Rezaie, H. Soltanain-Zadeh, Senior Member, IEEE, M.-R. Siadat, K.V. Elisevich, Medical Data Mining using Particle Swarm Optimization for Temporal Lobe Epilepsy, 2006 IEEE Congress on Evolutionary Computation.

[9] Agrawal, R. and Srikant, R. 1994, Fast algorithms for mining association rules, in Proceedings of 20th International Conference on Very Large Data Bases, 487-499.

[10] Han, J. and Pei, J. 2000, Mining frequent patterns by pattern-growth: methodology and implications, ACM SIGKDD Explorations Newsletter 2, 2, 14-20.

[11] Wang, C., Tjortjis, C., PRICES: An Efficient Algorithm for Mining Association Rules, Lecture Notes in Computer Science, Volume 3177, Jan 2004, Pages 352 – 358.

[12] Yuan, Y., Huang, T., A Matrix Algorithm for Mining Association Rules, Lecture Notes in Computer Science, Volume 3644, Sep 2005, Pages 370 – 379.

[13] Parthasarathy, S., Efficient Progressive Sampling for Association Rules, ICDM 2002: 354-361.

[14] Chuang, K., Chen, M., Yang, W., Progressive Sampling for Association Rules Based on Sampling Error Estimation, Lecture Notes in Computer Science, Volume 3518, Jun 2005, Pages 505 – 515.

[15] Li, Y., Gopalan, R., Effective Sampling for Mining Association Rules, Lecture Notes in Computer Science, Volume 3339, Jan 2004, Pages 391 – 401.

[16] Tien Dung Do, Siu Cheung Hui, Alvis Fong, Mining Frequent Itemsets with Category-Based Constraints, Lecture Notes in Computer Science, Volume 2843, 2003, pp. 76 – 86.

[17] Tseng, M., Lin, W., Jeng, R., Maintenance of Generalized Association Rules Under Transaction Update and Taxonomy Evolution, Lecture Notes in Computer Science, Volume 3589, Sep 2005, Pages 336 – 345.

[18] Manning, A., Keane, J., Data Allocation Algorithm for Parallel Association Rule Discovery, Lecture Notes in Computer Science, Volume 2035, Page 413-420.

[19] Schuster, A. and Wolff, R. (2001), Communication-efficient distributed mining of association rules, in `Proc. of the 2001 ACM SIGMOD Int'l. Conference on Management of Data', Santa Barbara, California, pp. 473-484.

[20] Cheung, D., Han, J., Ng, V., Fu, A. and Fu, Y. (1996), A fast distributed algorithm for mining association rules, in `Proc. of 1996 Int'l. Conf. on Parallel and Distributed Information Systems', Miami Beach, Florida, pp. 31 - 44.

[21] Dorigo M, Bonaneau E, Theraulaz G (2000), Ant algorithms and stigmergy, Future Generation Computer Systems, 16, 851- 871.

[22] Eberhart RC, Shi Y (2001) Particle swarm optimization: developments, applications and resources, in Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Seoul, Korea.

[23] Toksari MD (2006), Ant colony optimization for finding the global minimum, Applied Mathematics and Computation, pp 308-316.

[24] Margarita Sordo, Shawn N. Murphy, Harvard Medical School, Charlestown, MA. USA, A PSO/ACO Approach to Knowledge Discovery in a Pharmacovigilance Context.

[25] Carlos Ordonez , Comparing Association Rules and Decision Trees for Disease Prediction, ACM HIKM'06, November 11, 2006, Arlington, Virginia, USA.

Author Profile

Varinder Singh received Masters Degrees in Information Technology from Lovely Professional University in 2013. He is now Assistant Professor (Dept. of Computer Science) in BAM Khalsa College Garhshankar Punjab, India.