Increasing Performance of Rule Mining in the Medical Domain Using Natural Intelligence Concepts

Veenu Mangat

Abstract—**This paper discusses how concepts derived from nature can be applied successfully to improve the performance of the rule mining process. These concepts are derived from swarm intelligence and behavior of frogs. Swarm Intelligence (SI) is the property of a system whereby the collective behavior of agents interacting locally with their environment causes coherent functional global patterns to emerge. SI provides a basis with which it is possible to explore collective (or distributed) problem solving without centralized control or the provision of a global model. 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 which makes them a suitable approach for representing real world medical knowledge. In this paper, two new algorithms for rule mining have been implemented and their performance has been evaluated over a medical database. Results show that the usability of the rules thus uncovered, is high in the medical domain, and it can be further improved by refining the fitness function. Section I discusses the basic concepts of rule mining and swarm intelligence. Section II describes conventional rule mining techniques and states the motivation behind using swarm intelligence and frog leaping for rule mining and classification. Section III describes the various algorithms that have been implemented in our study. Section IV describes the details of the experiment. Section V presents the results of the practical experiment followed by conclusions and future scope in section VI.**

Keywords- Fitness function, Particle Swarm Optimization, rule mining, rule quality, Shuffled Frog Leaping, Swarm intelligence

I. 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 is the process of extracting interesting correlations, frequent patterns and associations among items in data repositories.

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.

They also have advantages of simplicity, uniformity, transparency, and ease of inference, which makes them a suitable approach 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 [5].

Association rules have been used by researchers 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, to determine candidates for temporal lobe surgery, 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. This happens because the frequency requirement for rules is lowered for medical data. 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 [1]. Some other issues in medical data [2] [3] 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.

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 [4].

Another algorithm that takes its inspiration from nature, the SFL algorithm has the ability to perform a flexible robust search for a good combination of terms (logical conditions) involving values of the predictor attributes. Therefore, SFL has been modified and developed to suit our application requirement.

II. Traditional Rule Mining Approaches

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

Manuscript received July 3, 2010. The author is an Assistant Professor at University Institute of Engineering and Technology, Panjab University, Chandigarh, India (email:veenumangat@yahoo.com).

subsets of I and X ∩ 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 which can function as a class label, 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 strength of the association rules

Rule mining problem is usually decomposed into two subproblems. One is to find those itemsets whose occurrences exceed a predefined threshold in the database; those itemsets are called frequent or large itemsets. The second problem is to generate association rules from those large itemsets with the constraints of minimal 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 pass 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 [6] style algorithm or FP-tree [7] 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 [8] [9], or ii) sampling data [10] [11] [12], or iii) adding extra constraints on the structure of rules [13] [14] or iv) parallelization of operations [15] [16] [17] 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.

Currently, meta-heuristic algorithms mainly include Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). These meta-heuristic algorithms have their respective strengths and weaknesses in rule mining. For instance, GA may take too long in searching for the optimal solution while ACO, involving quite a number of parameters, is easy to fall in local optimum. PSO is relatively simple and has evoked interest of researchers in different areas.

A combination of ACO/PSO algorithms has been proven to give acceptable results [25]. Also, a new meta-heuristic algorithm, Shuffled Frog-Leaping, works through observing, imitating and modeling the behavior of frogs who search for food laid on discrete stones randomly located in a pond. The shuffled frog-leaping algorithm draws its formulation from two other search techniques: the local search of the particle swarm optimization technique and the competitiveness mixing of information of the shuffled complex evolution technique. These latter two algorithms are the subject of our experiment.

III. Rule Mining using Swarm Intelligence

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 [18]. The autonomous agent does not follow commands from a leader, or some global plan [19].

Particle Swarm Optimization (PSO) and Ant Colonies Optimization (ACO) are currently the most popular algorithms in the swarm intelligence domain. In addition, the shuffled frog leaping (SFL) algorithm [20] has emerged recently as a new meta-heuristic derived from nature.

A. Combined ACO/PSO

This algorithm 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:

$$\tau_{cij} = \tau_{cij} + (\varphi_1 * Q_c) \text{ , for all } ij \text{ belongs to } \mathit{CurrentRule} \tag{1}$$

$$\tau_{cij} = \tau_{cij} + (\varphi_2 * Q_P) \text{ , for all } ij \text{ belongs to } \mathit{BestPastRule} \tag{2}$$

$$\tau_{cij} = \tau_{cij} + (\varphi_3 * Q_l) \text{ , for all } ij \text{ belongs to } \mathit{BestLocalRule} \tag{3}$$

$$\tau_{cij} = \tau_{cij} \,/\, \left(\Sigma^{ai+1}{}_{j=1} \, \tau_{cij} \right) \tag{4}$$

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 (5). φ is a random learning factor in the range 0..1.

$$Q=(TruePos/(TruePos+FalseNeg))*(TrueNeg/(FalsePos+TrueNeg)) \tag{5}$$

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 [21].

B. *ACO/PSO with Precision Fitness*

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 NRalgorithm 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 U 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 (6)

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

$$x_{id}=x_{id}+ v_{id} \tag{6}$$

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 (7):

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

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

ELSE *Q=Laplace-Corrected Precision* (7)

where *MinTP* is the least number of correctly covered examples that a rule has to cover [22].

C. *Shuffled Frog Leaping(SFL)*

Shuffled frog-leaping algorithm (SFL) is a new memetic meta-heuristic algorithm with efficient mathematical function and global search capability. It involves a set of frogs that cooperate with each other to achieve a unified behavior for the system as a whole, producing a robust system capable of finding high quality solutions for problems with a large search space. The pseudocode of SFL algorithm [20][23] is as follows:

```
Begin;
Generate random population of $P$ solutions (frogs);
For each individual $i$ that belongs to $P$ : calculate fitness (i);
Sort the population $P$ in descending order of their fitness;
Divide $P$ into $m$ memeplexes;
For each memeplex;
Determine the best and worst frogs;
Improve the worst frog position using (8);
$\text{NewpositionX}_i = \text{CurrentpositionX}_i + \text{New}\textit{Velocity}\text{V}_i$          (8)
Repeat for a specific number of iterations;
End;
Combine the evolved memeplexes;
Sort the population $P$ in descending order of their fitness;
Check if termination=true;
End;
```

An initial population of *P* frogs is created randomly. The structure of an individual for rule mining problem is composed of a set of attribute values. The velocity of individual *i* corresponds to the attribute update quantity covering all attribute values, the velocity of each individual is also created at random. The elements of position and velocity have the same dimension. In the next step, the frogs are sorted in a descending order according to their fitness. Fitness is defined as given by (5). The entire population is divided into *m* memeplexes, each containing n frogs. In this process, the first frog goes to the first memeplex, the second frog goes to the second memeplex, frog m goes to the *m*th memeplex, and frog m+1 goes back to the first memeplex, etc. Within each memeplex, a process similar to PSO is applied to improve only the frog with the worst fitness (not all frogs) in each cycle. If no improvement becomes possible in this case, then a new solution is randomly generated to replace that frog. The calculations then continue for a specific number of iterations. Rule pruning is done iteratively to remove one-term at a time from the rule while this process improves the quality of the rule, and the quality of the resulting rule is computed by (5).

D. *SFL with Precision Fitness*

As discussed in the previous section, the SFL algorithm has the ability to perform a flexible robust search for a good combination of terms (logical conditions) involving values of the various attributes. To prevent local optima, a submemeplex is constructed in each memeplex, which consists of frogs chosen on the basis of their respective fitness. The better the fitness, the easier it is chosen. So we modify the fitness function to suit our application requirements, which is that the false positives should be penalized severely. Quality (in turn fitness) is now defined using (7). The rest of the algorithm is the SFL discussed in previous subsection.

IV. Experimental Setup

A. *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 [24]. The primary data is sourced from 'entry' table of database. Our study focuses on identifying the relationship between alcohol intake (9 attributes), smoking (3 attributes), activities (4 attributes) and biochemical attributes (3 attributes). These can be used to classify a patient as hyperlipidemic or not. Hyperlipidemia is defined as the presence of high levels of cholesterol and/or triglycerides in the blood. It is not a disease but a metabolic derangement that can be secondary to many diseases and can contribute to many forms of disease, most notably cardiovascular disease. Data corresponding to 1419 persons has been considered. The attributes were manually extracted. Continuous attributes need to be 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.

B. *Setting of Parameters*

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 (7) 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. As SFLA is a relatively new algorithm, there is no theoretical basis for parameters setting. The number of memeplexes has been taken as 10, frog population is set to 10*n* (*n* being number of attributes), number of submemeplexes is set to 2*n*/3, number of independent runs was 30 and local exploration was carried out *n* times.

V. 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 I summarizes the results obtained by the combined ACO/PSO, ACO/PSO with Precision Fitness, SFL and SFL with Precision Fitness algorithms.

TABLE I. COMPARISON BETWEEN ACO/PSO, ACO/PSO WITH PF, SFL, SFL WITH PF

	Accuracy	**No. of rules in rule set**	**No. of conditions in rule**
ACO/PSO	85.1%	17.1± 2.27	9.88± 0.21
ACO/PSO with PF	91.5%	13.3± 1.0	6.81± 0.14
SFL	90.1%	8.2	5.3
SFL with PF	94.2%	6.1	4.4

VI. CONCLUSIONS AND FUTURE WORK

The rule quality can be viewed in terms of its accuracy and comprehensibility. A rule will be usable by a medical practitioner if it is accurate and easily understood. All four techniques studied provide accuracy comparable to other non SI based mining approaches. SFL with PF shows very good results. 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 interesting for medical knowledge discovery. Shuffled Frog Leaping 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. To balance between efficiency and exploration capability, extensive experiments need to be conducted with different settings of parameters to arrive at the optimal values for these algorithms.

REFERENCES

[1] 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.

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

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

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

[6] 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.

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

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

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

[10] Parthasarathy, S., “Efficient Progressive Sampling for Association Rules”, *ICDM 2002*: 354-361.

[11] 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.

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

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

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

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

[20] M.M. Eusuff, K.E.Lansey, “Optimization of water distribution network design using the shuffled frog leaping algorithm”, *Journal of Water Resources Planning and Management*, American Society of Civil Engineers, Reston, June 2003,pp. 210-225.

[21] N. Holden and A. A. Freitas, “A hybrid PSO/ACO algorithm for classification”, in *Proceedings of the 9th Genetic and Evolutionary Computation Conference Workshop on Particle Swarms: The Second Decade (GECCO ’07)*, pp. 2745–2750, ACM Press, London, UK, July 2007.

[22] Holden, N., Freitas, A. “A Hybrid PSO/ACO Algorithm for Discovering Classification Rules in Data Mining”, In *Journal of Artificial Evolution and Applications (JAEA)*, 2008.

[23] E.Elbeltagi, T.Hegazy, and D.Grierson, “Comparison among five evolutionary-based optimization algorithms”, *Advanced Engineering Informatics, Elsevier Science Publishers*, Amsterdam, January 2005, pp.43-53.

[24] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, CRISP-DM 1.0. Technical report, *The CRISP-DM Consortium*, 2000.

[25] Veenu Mangat, “Swarm Intelligence Based Technique for Rule Mining in the Medical Domain”, *International Journal of Computer Applications IJCA*, July, 2010.