Genetic Algorithms for Multi-Criterion Classification and Clustering in Data Mining
Satchidananda Dehuri, Ashish Ghosh and Rajib Mall
Pages 143 – 154

Genetic Algorithms for Multi-Criterion Classification and Clustering in Data Mining

Satchidananda Dehuri
Department of Information & Communication Technology
Fakir Mohan University,
Vyasa Vihar, Balasore-756019, India
Email: satchi_d@yahoo.co.in

Ashish Ghosh
Machine Intelligence Unit and Center for Soft Computing Research
Indian Statistical Institute,
203, B.T. Road, Kolkata–700108, INDIA.
Email: ash@isical.ac.in

Rajib Mall
Department of Computer Science & Engineering
Indian Institute of Technology,
Kharagpur-721302, India.
Email: rajib@cse.iitkgp.ernet.in

Abstract: *This paper focuses on multi-criteria tasks such as classification and clustering in the context of data mining. The cost functions like rule interestingness, predictive accuracy and comprehensibility associated with rule mining tasks can be treated as multiple objectives. Similarly, complementary measures like compactness and connectedness of clusters are treated as two objectives for cluster analysis. We have carried out an extensive simulation for these tasks using different real life and artificially created datasets. Experimental results presented here show that multi-objective genetic algorithms (MOGA) bring a clear edge over the single objective ones in the case of classification task; whereas for clustering task they produce comparable results.*

Keywords : *MOGA, Data Mining, Classification, Clustering.*

Received: *November 05, 2005* | **Revised:** *April 15, 2006* | **Accepted:** *June 12, 2006*

1. Introduction

The commercial and research interests in data mining is increasing rapidly, as the amount of data generated and stored in databases of organizations is already enormous and continuing to grow very fast. This large amount of stored data normally contains valuable hidden knowledge, which, if harnessed, could be used to improve the decision making process of an organization. For instance, data about previous sales might contain interesting relationships between products, types of customers and buying habits of customers. The discovery of such relationships can be very useful to efficiently manage the sales of a company. However, the volume of the archival data often exceeds several gigabytes or even terabytes, which is beyond the analyzing capability of human beings. Thus there is a clear need for developing semi-automatic methods for extracting knowledge from data.

Traditional statistical data summarization, database management techniques and pattern recognition techniques are not adequate for handling data of this scale. This quest led to the emergence of a field called data mining and knowledge discovery (KDD) [1] aimed at discovering natural structures/ knowledge/hidden patterns within such massive data. Data mining (DM), the core step of KDD, deals with the process of identifying valid, novel and potentially useful, and ultimately understandable patterns in data. It involves the following tasks: classification, clustering, association rule mining, sequential pattern analysis and data visualization [3, 4].

In this paper we are considering classification and clustering. Each of these tasks involves many criteria. For example, the task of classification rule mining involves the measures such as comprehensibility, predictive accuracy, and interestingness [5]; and the task of clustering involves compactness as well as connectedness of clusters [6]. In this work, we tried to solve these tasks by multi-objective genetic algorithms [7], thereby removing some of the limitations of the existing single objective based approaches.

The remainder of the paper is organized as follows: In Section 2, an overview of DM and KDD process is presented. Section 3 presents a brief survey on the role of genetic algorithm for data mining tasks. Section 4 presents the new dimension to data mining and KDD using MOGA. In Section 5 we give the experimental results with analysis. Section 6 concludes the article.

2. An Overview of DM and KDD

Knowledge discovery in databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [1]. It is interactive and iterative, involving numerous steps with many decisions being made by the user.

Here we mention that the discovered knowledge should have three general properties: namely, predictive accuracy, understandability, and interestingness in the parlance of classification [8]. Properties like compactness and connectedness are embedded in clusters. Let us briefly discuss each of these properties.

- *Predictive Accuracy*: The basic idea is to predict the value that some attribute(s) will take in "future" based on previously observed data. We want the discovered knowledge to have a high predictive accuracy.

- *Understandability*: We also want the discovered knowledge to be comprehensible for the user. This is necessary whenever the discovered knowledge is to be used for supporting a decision to be made by a human being. If the discovered knowledge is just a black box, which makes predictions without explaining them, the user may not trust it [9]. Knowledge comprehensibility can be achieved by using high-level knowledge representations. A popular one in the context of data mining, is a set of IF- THEN (prediction) rules, where each rule is of the form

  $$If \ < antecedent > then < consequent >.$$

  If the number of attributes is small for the antecedent as well as for the consequent clause, then the discovered knowledge is understandable.

- *Interestingness*: This is the third and most difficult property to define and quantify. However, there are some aspects of knowledge interestingness that can be defined in objective ways. The topic of rule interestingness, including a comparison between the subjective and the objective approaches for measuring rule interestingness, will be discussed in Section 3; and interested reader can refer to [10] for more details.

- *Compactness*: To measure the compactness of a cluster we compute the overall deviation of a partitioning. This is computed as the overall sum of square distances for the data items from their corresponding cluster centers. Overall deviation should be minimized.

- *Connectedness*: The connectedness of a cluster is measured by the degree to which neighboring data points have been placed in the same clusters. As an objective, connectivity should be minimized. The details of these two objectives related to cluster analysis is discussed in Section 5.

2.2 Data Mining

Data mining is one of the important steps of KDD process. The common algorithms in current data mining practice include the following.

1) *Classification*: classifies a data item into one of several predefined categories /classes.
2) *Regression*: maps a data item to a real-valued prediction variable.
3) *Clustering*: maps a data item into one of several clusters, where clusters are natural groupings of data items based on similarity matrices or probability density models.
4) *Discovering association rules*: describes association relationship among different attributes.
5) *Summarization*: provides a compact description for a subset of data.
6) *Dependency modeling*: describes significant dependencies among variables.
7) *Sequence analysis*: models sequential patterns like time-series analysis. The goal is to model the states of the process generating the sequence or to extract and report deviation and trends over time.

Since in the present article we are interested in the following two important tasks of data mining, namely classification and clustering; we briefly describe them here.

Classification: This task has been studied for many decades by the machine learning and statistics communities [11]. In this task the goal is to predict the value (the class) of a user specified goal attribute based on the values of other attributes, called

predicting attributes. Classification rules can be considered as a particular kind of prediction rules where the rule antecedent ("IF" part) contains predicting attribute and rule consequent ("THEN" part) contains a predicted value for the goal attribute. An example of classification rule is:

IF (Attendance > 75%) and (total_marks >60%) THEN (result= "pass").

In the classification task the data being mined is divided into two mutually exclusive and exhaustive sets, the training set and the test set. The DM algorithm has to discover rules by accessing the training set; and. the predictive performance of these rules is evaluated on the test set (not seen during training). A measure of predictive accuracy is discussed in a later section; the reader may refer to [12] also.

Clustering: In contrast to classification task, in the clustering process the data-mining algorithm must, in some sense, discover the classes by partitioning the data set into clusters, which is a form of unsupervised learning [13]. Examples that are similar to each other tend to be assigned to the same cluster, whereas examples different from each other belong to different clusters. Applications of GAs for clustering are discussed in [14,15].

3. GA Based DM Tasks

This section is divided into two parts. Subsection 3.1, discusses the use of genetic algorithms for classificatory rule generation, and Subsection 3.2 discusses the use of genetic algorithm for data clustering.

3.1 Genetic Algorithms (GAs) for Classification

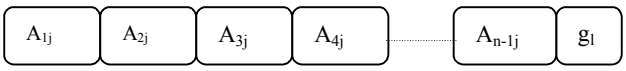
Genetic algorithms are probabilistic search algorithms. At each steps of such algorithm a set of N potential solutions (called individuals $I_k \in \Xi$, where Ξ represents the space of all possible individuals) is chosen in an attempt to describe as good as possible solution of the optimization problem [19,20,21]. This population P= $\{I_1, I_2, \ldots I_N\}$ is modified according to the natural evolutionary process. After initialization, selection S: $I^N \rightarrow I^N$ and recombination $Я : I^N \rightarrow I^N$ are executed in a loop until some termination criterion is reached. Each run of the loop is called a generation and P (t) denotes the population at generation t.

The selection operator is intended to improve the average quality of the population by giving individuals of higher quality a higher probability to be copied into the next generation. Selection thereby focuses on the search of promising regions in the search space. The quality of an individual is measured by a fitness function f: $P \rightarrow$ R. Recombination changes the genetic material in the population either by crossover or by mutation in order to obtain new points in the search space.

3.1.1 Genetic Representations

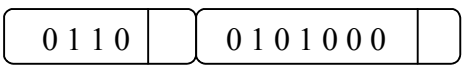
Each individual in the population represents a candidate rule of the form *"if Antecedent* then *Consequent"*. The antecedent of this rule can be formed by a conjunction of at most n – 1 attributes, where n is the number of attributes being mined. Each condition is of the form $A_i = V_{ij}$, where A_i is the i-th attribute and V_{ij} is the j-th value of the i-th attribute's domain. The consequent consists of a single condition of the form $G = g_l$, where G is the goal attribute and g_l is the l^{th} value of the goal attribute's domain.

A string of fixed size encodes an individual with n genes representing the values that each attribute can assume in the rule as shown below. In addition, each gene also contains a Boolean flag (f_p /f_a) except the n^{th} gene that indicates whether or not the i^{th} condition is present in the rule antecedent. Hence although all individuals have the same genome length, different individuals represent rules of different lengths.



Let us see how this encoding scheme is used to represent both categorical and continuous attributes present in the dataset. In the categorical (nominal) case, if a given attribute can take on k-discrete values then we can encode this attribute by using k-bits. The i^{th} value (i=1,2,3…,k) of the attribute's domain is a part of the rule if and only if i^{th} bit is 1.

For instance, suppose that a given individual represents two attribute values, where the attributes are branch and semester and their corresponding values can be EE, CS, IT, ET and 1^{st}, 2^{nd}, 3^{rd}, 4^{th}, 5^{th}, 6^{th}, 7^{th}, 8^{th} respectively. Then a condition involving these attributes would be encoded in the genome by four and 8 bits respectively. This can be represented as follows:



to be interpreted as

If (branch = CS or IT) and (semester=2^{nd} or 4^{th}).

Hence this encoding scheme allows the representation of conditions with internal disjunctions, i.e. with the logical 'OR' operator within a condition. Obviously this encoding scheme can be easily extended to represent rule antecedent with several conditions (linked by a logical AND).

In the case of continuous attributes the binary encoding mechanism gets slightly more complex. A common approach is to use bits to represent the value of a continuous attribute in binary notation. For instance the binary string 00001101 represents the value 13 of a given integer-value attributes.

Similarly the goal attribute is also encoded in the individual. This is one possibility. The second possibility is to associate all individuals of the population with the same predicted class, which is never modified during the execution of the algorithm. Hence if we want to discover a set of classification rules predicting 'k' different classes, we would need to run the evolutionary algorithm at least k-times, so that in the i^{th} run, i=1,2,3..,k, the algorithm discovers only rules predicting the i^{th} class [22].

3.1.2 Fitness Function

As discussed in Section 2.1, the discovered rules should have (a) high predictive accuracy (b) comprehensibility and (c) interestingness. In this subsection we discuss how these criteria can be defined and used in the fitness evaluation of individuals in GAs.

1.*Comprehensibility Metric*: There are various ways to quantitatively measure rule comprehensibility. A standard way of measuring comprehensibility is to count the number of rules and the number of conditions in these rules. If these numbers increase then comprehensibility decreases.

If a rule *R* can have at most M conditions, the comprehensibility of a rule *C(R)* can be defined as:

$$C(R) = M - (\text{number of condition } (R)). \quad (1)$$

2.*Predictive Accuracy*: As already mentioned, our rules are of the form IF A THEN C. The antecedent part of the rule is a conjunction of conditions. A very simple way to measure the predictive accuracy of a rule is

$$\text{Predic}Acc = \frac{|A \,\&\, C|}{|A|}, \quad (2)$$

where $|A \,\&\, C|$ is defined as the number of records satisfying both A and C.

3.*Interestingness*: The computation of the degree of interestingness of a rule, in turn, consists of two terms. One of them refers to the antecedent of the rule and the other to the consequent. The degree of interestingness of the rule antecedent is calculated by an information-theoretical measure, which is a normalized version of the measure proposed in [25, 26] defined as follows:

$$RInt = 1 - \frac{\sum_{i=1}^{n-1} InfoGain(A_i)}{\frac{n-1}{\log_2(|dom(G)|)}} \quad (3)$$

where 'n' is the number of attributes in the antecedent and $|dom(G)|$ is the domain cardinality (i.e. the number of possible values) of the goal attribute G occurring in the consequent. The log term is included in the formula (3) to normalize the value of RInt, so that this measure takes a value between 0 and 1. The InfoGain is given by:

$$InfoGain(A_i) = Info(G) - Info(G \mid A_i) \quad (4)$$

with

$$Info(G) = -\sum_{i=1}^{m_k} \left(P(g_l) \log_2(p(g_l))\right) \quad (5)$$

$$Info(G \mid A_i) = \sum_{i=1}^{n_i} \left(p(v_{ij}) \left(-\sum_{j=1}^{m_k} p(g_l \mid v_{ij}) \log_2(p(g_l \mid v_{ij})) \right) \right) \quad (6)$$

where m_k is the number of possible values of the goal attribute G_k, n_i is the number of possible values of the attribute A_i, *p(X)* denotes the probability of X and *p(X|Y)* denotes the conditional probability of X given Y.

The overall fitness is computed as the arithmetic weighted mean as

$$f(x) = \frac{w_1 \cdot C(R) + w_2 \cdot PredAcc + w_3 \cdot RInt}{w_1 + w_2 + w_3}, \quad (7)$$

where w_1, w_2 and w_3 are user-defined weights.

3.1.3 Genetic Operators

The crossover operator we consider here follows the idea of uniform crossover [27, 28]. After crossover is completed, the algorithm analyses if any invalid individual is created. If so, a repair operator is used to produce valid individuals.

The mutation operator randomly transforms the value of an attribute into another value belonging to the same domain of the attribute.

Besides crossover and mutation, the insert and remove operators directly try to control the size of the rules being evolved; thereby influence the comprehensibility of the rules. These operators randomly insert and remove, a condition in the rule antecedent. These operators are not part of the regular GA. However we have introduced them here for suitability in our rule generation scheme.

3.2 Genetic Algorithm for Data Clustering

A lot of research has been conducted on applying GAs to the problem of k clustering, where the required number of clusters is known [29]. Adaptation to the k-clustering problem requires individual representation, fitness function creation, operators, and parameter values.

3.2.1 Individual Representation

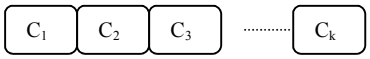
The classical ways of genetic representations for clustering or grouping problems are based on two underlying schemes. The first one allocates one (or more) integer or bits to each object, known as genes, and uses the values of these genes to signify which cluster the object belongs to. The second scheme represents the objects with gene values, and the positions of these genes signify how the objects are divided amongst the clusters. Figure 1 shows encoding of the clustering $\{\{O_1, O_2, O_4\}, \{O_3, O_5, O_6\}\}$ by group number and matrix representations, respectively.

Group-number encoding is based on the first encoding scheme and represents a clustering of n objects as a string of n integers where the i^{th} integer signifies the group number of the i^{th} object. When there are two clusters this can be reduced to a binary encoding scheme by using 0 and 1 as the group identifier.

Bezdek et al. [30] used $k \times n$ matrix to represent a clustering, with each row corresponding to a cluster and each column associated with an object. A 1 in row i, column j means that object j is in group i. Each column contains exactly one 1, whereas a row can have many 1's. All other elements are 0's. This representation can also be adapted for overlapping clusters or fuzzy clustering.

For the k-clustering problem, any chromosome that does not represent a clustering with k groups is necessarily invalid: a chromosome that does not include all group numbers as gene values is invalid; a matrix encoding with a row of 0's is invalid. A matrix encoding is also invalid if there is more than one 1 in any column. Chromosomes with group values that do not correspond to a group or object, and permutations with repeated or missing object identifiers are invalid.

Though these two representation schemes are easier but limitation arises if we represent a million of records, which are often encountered in data mining. Hence the present representation scheme uses an alternative approach proposed in [31]. Here each individual consists of k-cluster centers such as $C_1, C_2, C_3, \ldots C_K$. Center C_i represents the number of features of the available feature space. For an N-dimensional feature space the total length of the individual is $k \times n$ as shown below.



3.2.2 Fitness Function

Objective functions used for traditional clustering algorithms can act as fitness functions for GAs. However, if the optimal clustering corresponds to the minimal objective functional value, one needs to transform the objective functional value since GAs work to maximize the fitness values. In addition, fitness values in a GA need to be positive if we are using fitness proportional selection. Krovi [14] used the ratio of sum of squared distances between clusters and sum of squared distances within a cluster as the fitness function. Since the aim is to maximize this value, no transformation is necessary. Bhuyan et al, [32, 33] used the sum of squared Euclidean distance of each object from the centroid of its cluster for measuring fitness. This value is then transformed ($f' = C_{\max} - f$, where f is the raw fitness, f' is the scaled fitness, and C_{max} is the value of the poorest string in the population) and linearly scaled to get the fitness value. Alippi and Cucchiara [33] also used the same criterion, but used a GA that has been adapted to minimize fitness values. Bezdek et al.'s [30 clustering criterion is also based around minimizing the sum of squared distances of objects from their cluster centers, but they used three different distance metrics (Euclidean, diagonal, and Mahalanobis) to allow for different cluster shapes.

4.3 Genetic Operators

Selection

Chromosomes are selected for reproduction based on their relative fitness. If all the fitness values are positive, and the maximum fitness value corresponds to the optimal clustering, then fitness proportional selection may be appropriate. Otherwise, a ranking selection method may be used. In addition, elite selection will ensure that the fittest chromosomes are passed from one generation to the next. Krovi [14] used the fitness proportional selection [21]. The selection operator used by Bhuyan et al. [32] is an elitist version of fitness proportional selection. A new population is formed by picking up the x (a parameter provided by the user) better strings from the combination of the old population and offspring. The remaining chromosomes in the population are selected from the offspring.

Crossover

Crossover operator is designed to transfer genetic material from one generation to the next. Major concerns with this operator are validity and context insensitivity. It may be necessary to check whether offspring produced by a certain operator is valid.

Context insensitivity occurs when the crossover operator used in a redundant representation acts on the chromosomal level instead of the clustering level. In this case the child chromosome may resemble the parent chromosomes, but the child clustering does not resemble the parent clustering. Figure 2 shows that the single point crossover is context insensitive for group number representation.

Here both parents represent the same clustering, $\{\{O_1, O_2, O_3\}, \{O_4, O_5, O_6\}\}$ although the group

numbers are different. Given that the parents represent the same solution, we would expect the children to also represent this solution. Instead, both children represent the clustering $\{O_1, O_2, O_3, O_4, O_5, O_6\}$ which does not resemble either of the parents.

The crossover operator for matrix representation is as follows:

Alippi and Cucchiara [33] used a single–point asexual crossover to avoid the problem of redundancy (Figure 3). The tails of two rows of the matrix are swapped, starting from a randomly selected crossover point. This operator may produce clustering with less than 'k' groups.

Bezdek et al. [30] used a sexual 2-point crossover (Figure 4). A crossover point and a distance (the number of columns to be swapped) are randomly selected–these determine which columns are swapped between the parents. This operator is context insensitive and may produce offspring with less than k groups.

Mutation

Mutation introduces new genetic material into the population. In a clustering context this corresponds to moving an object from one cluster to another. How this is done is dependent on the representation.

Group number

Krovi [14] used the mutation function implemented by Goldberg [21]. Here each bit of the chromosome is inverted with a probability equal to the mutation rate, p_{mut}. Jones and Beltramo [33] changed each group number (provided it is not the only object left in that group) with probability, $p_{mut} = 1/n$ where n is the number of objects.

Matrix

Alippi and Cucchiara [33] used a column mutation, which is shown in Figure 5. An element is selected from the matrix at random and set to 1. All other elements in the column are set to 0. If the selected element is already 1 this operator has no effect. Bezdek et al. [30] also used a column matrix, but they chose an element and flipped it.

4. Multi-Criteria Optimization by GAs

4.1 Multi-criteria optimization

Multi-objective optimization methods deal with finding optimal (!) solutions to problems having multiple objectives [34, 35]. Thus for this type of problems the user is never satisfied by finding one solution that is optimum with respect to a single criterion. The principle of a multi-criteria optimization procedure is different from that of a single criterion optimization. In a single criterion optimization the main goal is to find the global optimal solutions. However, in a multi-criteria optimization problem, there is more than one objective function, each of which may have a different individual optimal solution. If there is a sufficient difference in the optimal solutions corresponding to different objectives then we say that the objective functions are conflicting. Multi-criteria optimization with such conflicting objective functions gives rise to a set of optimal solutions, instead of one optimal solution known as Pareto-optimal solutions [36].

Let us illustrate the Pareto optimal solution with time & space complexity of an algorithm shown in Figure 6. In this problem we have to minimize both times as well as space requirements. The point 'p' represents a solution, which has minimal time but high space complexity. On the other hand, the point 'r' represents a solution with high time complexity but minimum space complexity. Considering both the objectives, no solution is optimal. So in this case we can't say that solution 'p' is better than 'r'. In fact, there exists many such solutions like 'q' that belong to the *Pareto optimal set* and one can't sort the solution according to the performance metrics considering both the objectives. All the solutions, on the curve, are known as *Pareto-optimal solutions*. From Figure-6 it is clear that there exists solutions like 't', which do not belong to the Pareto optimal set.

Let us consider a problem having m objectives (say $f_i, i = 1,2,3,.....,m$ and $m > 1$). Any two solutions $u^{(1)}$ and $u^{(2)}$ (having't' decision variables each) can have one of two possibilities-one dominates the other or none dominates the other. A solution $u^{(1)}$ is said to *dominate* the other solution $u^{(2)}$, if the following conditions are true:

1. The solution $u^{(1)}$ is not worse (say the operator $\prec$ denotes worse and $\succ$ denotes better) than $u^{(2)}$ in all objectives, or $f_i(u^{(1)}) \geq f_i(u^{(2)}), \forall i = 1,2,3....,m.$

2. The solution $u^{(1)}$ is strictly better than $u^{(2)}$ in at least one objective, or $f_i(u^{(1)}) \succ f_i(u^{(2)})$ for at least one, $i \in \{1,2,3,..,m\}$.

If any of the above conditions is violated, the solution $u^{(1)}$ does not dominate the solution $u^{(2)}$. If $u^{(1)}$ dominates the solution $u^{(2)}$, then we can also say that $u^{(2)}$ is dominated by $u^{(1)}$, or $u^{(1)}$ is non-dominated

by $u^{(2)}$, or simply between the two solutions, $u^{(1)}$ is the non-dominated solution.
Multi-criterion optimization algorithms try to achieve mainly the following two goals:
1.Guide the search towards the global Pareto-optimal region, and
2.Maintain population diversity in the Pareto-optimal front.
The first task is a natural goal of any optimization algorithm. The second task is unique to multi-criterion optimization.

Multi-criterion optimization is not a new field of research and application in the context of classical optimization. The weighted sum approach [37], ε-perturbation method [37], goal programming [38], Tchybeshev method [38], min-max method [38] and others are all popular methods often used in practice [39]. The core of these algorithms, is a classical optimizer, which can at best, find a single optimal solution in one simulation. In solving multi-criterion optimization problems, they have to be used many times, hopefully finding a different Pareto-optimal solution each time. Moreover, these classical methods have difficulties with problems having non-convex search spaces.

4.2 Multi-criteria GAs

Evolutionary algorithms (EAs) are a natural choice for solving multi-criterion optimization problems because of their population-based nature. A number of Pareto-optimal solutions can, in principle, be captured in an EA population, thereby allowing a user to find multiple Pareto-optimal solutions in one simulation. The fundamental difference between a single objective and multi-objective GA is that in the single objective case fitness of an individual is defined using only one objective, whereas in the second case fitness is defined incorporating the influence of all the objectives. Other genetic operators like selection and reproduction are similar in both cases. The possibility of using EAs to solve multi-objective optimization problems was proposed in the seventies. David Schaffer was the first to implement Vector Evaluated Genetic Algorithm (VEGA) [35] in the year 1984. There was lukewarm interest for a decade, but the major popularity of the field began in 1993 following a suggestion by David Goldberg based on the use of the non-domination [21] concept and a diversity- preserving mechanism. There are various multi-criteria EAs proposed so far, by different authors and good surveys are available in [40, 41].

For our task we shall use the following algorithm.

Algorithm

1. *g=1; External (g)=ϕ;*
2. *Initialize Population P(g);*
3. *Evaluate the P(g) by Objective Functions;*
4. *Assign Fitness to P(g) Using Rank Based on Pareto Dominance*
5. *External (g) ← Chromosomes Ranked as 1;*
6. *While (g <= Specified_no_of_Generation) do*
7. *P'(g)← Selection by Roulette Wheel Selection Schemes P(g);*
8. *P"(g)← Single-Point Uniform Crossover and Mutation P'(g);*
9. *P"'(g)← Insert/Remove Operation P"(g);*
10. *P(g+1)← Replace (P(g), P"'(g));*
11. *Evaluate P(g+1) by Objective Functions;*
12. *Assign Fitness to P(g+1) Using Rank Based Pareto Dominance;*
13. *External (g+1) ← [External (g) + Chromosome Ranked as One of P(g+1)];*
14. *g=g+1;*
15. *End while*
16. *Decode the Chromosomes Stored in External as an IF-THEN Rule*

5. MOGA for DM tasks

5.1 MOGA for classification

As stated in Section-2, classification task has many criteria such as predictive accuracy, comprehensibility, and interestingness. These three are treated as multiple objectives of our mining scheme. Let the symbols f_1, f_2, and f_3 correspond to predictive accuracy; comprehensibility and rule interestingness (need to be maximized).

5.1.1 Experimental Details

Description of the Dataset

Simulation was performed using benchmark the zoo and nursery dataset obtained from the UCI machine repository (http://www.ics.uci.edu/).

Zoo Data
The zoo dataset contains 101 instances and 18 attributes. Each instance corresponds to an animal. In the preprocessing phase the attribute containing the name of the animal was removed. The attributes are all categorical, namely hair(h), feathers(f), eggs(e), milk(m), predator(p), toothed(t), domestic(d), backbone(b), fins(fs), legs(l), tail(tl), catsize(c), airborne(a), aquatic(aq), breathes(br), venomous(v) and type(ty). Except type and legs, all other attributes are Boolean. The goal attributes are type 1 to 7. The type 1 has 41 records, type 2 has 20 records, type 3 has 5 records, type 4, 5, 6, & 7 has 13, 4, 8, 10 records respectively.

Nursery Data

This dataset has 12960 records and nine attributes having categorical values. The ninth attributes is treated as class attribute and there are five classes: not_recom (NR), recommended (R), very_recom (VR), priority (P), and spec_prior(SP). The attributes and corresponding values are listed in Table 1.

Results

Experiments have been performed using MATLAB 5.3 on a Linux server. The following parameters are used shown in Table 2.

P: population size
P_c : Probability of crossover
P_m : probability of mutation
R_m : Removal operator
R_I : Insert Operator

For each of the datasets the simple genetic algorithm had 100 individuals in the population and was run for 500 generations. The parameter values such as P_c, P_m, R_m, and R_i were sufficient to find some good individuals. The following computational protocols are used in the basic simple genetic algorithm as well as the proposed multi-objective genetic algorithm for rule generation. The data set is divided into two parts: training set and test set. Here we have used 30% for training set and the rest are test set. We represent the predicted class to all individuals of the population, which is never modified during the running of the algorithm. Hence, for each class we run the algorithms separately and get the corresponding rules.

Rules generated by MOGA have been compared with those of SGA and all rules are listed in the following table. Table 3 and 4 show the results generated by SGA and, MOGA respectively from zoo dataset. Table 3 has three columns namely class#, mined rules, and fitness value. Similarly, Table 4 has five columns which includes class#, mined rules, predictive accuracy, comprehensibility and interestingness measures.

Tables 5 and 6 show the result generated by SGA and MOGA respectively from nursery dataset. Table 5 has three columns namely class#, mined rules, and fitness value. Similarly, Table 6 has five columns which includes class#, mined rules, predictive accuracy, comprehensibility and interestingness measures.

5.2 MOGA for Clustering

Conventional genetic algorithm based data clustering utilize a single criterion that may not confirm to the diverse shapes of the underlying data. This section provides a novel approach to data clustering based on the explicit optimization of a partitioning with respect to multiple complementary clustering objectives [5]. It has been shown that this approach may be more robust to the variety of cluster structures found in different data sets, and may be able to identify certain cluster structures that cannot be discovered by other methods. MOGA for data clustering uses two complementary objectives based on cluster compactness and connectedness. Let us define the objective functions separately.

Compactness

Cluster compactness can be measured by the overall deviation of a partitioning. This is simply computed as the overall summed distances between data items and their corresponding cluster centers as

$$comp\ (S) = \sum_{c_k \in Si \in c_k} \sum d(i, \mu_k) \qquad (12)$$

where S is the set of all clusters, μ_k is the centroid of cluster c_k and d(..) is the choosen distance function (e.g. Euclidean distance). As an objective, overall deviation should be minimized. This criterion is similar to the popular criterion of intra-cluster variance, which squares the distance value d(..) and is more strongly biased towards spherically shaped clusters.

Connectedness

This measure evaluates the degree to which neighboring data points have been placed in the same cluster. It is computed as

$$Conn\ (S) = \sum_{i=1}^{N} \left(\sum_{j=1}^{L} x_{i,nn_i(j)} \right) \qquad (13)$$

$$\text{where} \quad x_{r,s} = \begin{cases} \frac{1}{j} if\ \ni c_k : r,s \in c_k \\ 0 \quad otherwise \end{cases}$$

$nn_i(j)$ is the j^{th} nearest neighbor of datum i and L is a parameter determining the number of neighbors that contributes to the connectivity measure. As an objective, connectivity should be minimized. After defining theses two objectives, then the algorithms that are defined in Section 4.2 can be applied to optimize them simultaneously. The genetic operators such as crossover, mutation is the same as single objective genetic algorithm for data clustering.

5.2.1 Experimental Details

Parameters taken for simulations are $0.6 \le \mu_c \le 0.8$ and $0.001 \le \mu_m \le 0.01$. We have carried out extensive simulation using labeled data sets for easy validation of our results. Table 7 shows the results obtained from both SGA based clustering and proposed MOGA based clustering.

Population size was taken as 200. Other parameters like selection, crossover and mutation were used for the simulation. MOGA based clustering generate solutions that are comparable or better than the simple genetic algorithm. In the case of IRIS data set

both the connectivity and compactness achieved a near optimal solution, whereas in the other two datasets named as wine and WBCD the results of both the objectives were very much conflicting to each other.

As expected the computational time requirement for MOGA is higher than the single objective based ones.

6. Conclusions and Discussion

In this paper we have discussed the use of multi-objective genetic algorithms for classification and clustering. In clustering, it has been demonstrated that MOGA based clustering shows robustness over the existing single objective ones. Finding more objectives that are hidden in cluster analysis as well as without using apriori knowledge of k-clusters is a promising research direction. The scalability, which is encountered in MOGA based rule mining from large databases/ data warehouses, is another major research area. Though MOGA is discussed for two tasks of data mining, it can be extended to the task like sequential pattern analysis and data visualization of data mining.

ACKNOWLEDGMENT

Dr. S. Dehuri is grateful to the *Center for Soft Computing Research, Indian Statistical Institute, Kolkata* for providing a fellowship to carry out this work.

REFERENCES

[1] U. M. Fayyad, G. Piatetsky-Shapiro and P. Smyth (1996). *From Data Mining to Knowledge Discovery: an Overview*. In: U.M. Fayyad, G. Piatetsky–Shapiro, P. Smyth and R. Uthurusamy (eds.), Advances in Knowledge Discovery & Data Mining, pp.1-34, AAAI/MIT.

[2] R. J. Brachman and T. Anand (1996). *The Process of Knowledge discovery in Databases: A Human Centered Approach.* In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, (eds.), Advances in knowledge Discovery and Data Mining, Chapter 2, pp. 37-57, AAAI/MIT Press.

[3] W. Frawley, G. Piatasky-Saprio, C. Mathews (1991). *Knowledge Discovery in Databases: An Overview*. AAAI/ MIT Press.

[4] J. Han, M. Kamber (2001). *Data Mining-Concepts and Techniques*. Morgan Kaufmann.

[5] A. A. Freitas (2002). *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer-Verlag, New York.

[6] J. Handl, J. Knowles (2004). *Multi-objective Clustering with Automatic Determination of the Number of Clusters*. Tech. Report TR-COMPSYSBIO-2004-02 UMIST, Manchester.

[7] A. Ghosh, B. Nath (2004). Multi-objective Rule Mining Using Genetic Algorithms. *Information Sciences*, Vol. 163, pp. 123-133.

[8] S. Dehuri, R. Mall (2004). *Mining Predictive and Comprehensible Classification Rules Using Multi-Objective Genetic Algorithm*. In Proceeding of ADCOM, pp. 99-104, India.

[9] D. Michie, D.J. Spiegelhalter, and C.C. Taylor (1994). *Machine Learning, Neural and Statistical Classification*. New York: Ellis Horwood.

[10] A.A. Freitas (1999). *On Rule Interestingness Measures*. Knowledge Based Systems, vol. 12, Pp 309-315.

[11] T.-S. Lim, W.-Y. Loh and Y.-S. Shih (2000). *A Comparison of Prediction Accuracy, Complexity and Training Time of Thirty-Three Old and New Classification Algorithms*. Machine Learning Journal, Vol. 40, pp. 203-228.

[12] J. D. Kelly Jr. and L. Davis (1991). *A Hybrid Genetic Algorithm for Classification*. Proc. 12th Int. Joint Conf. On AI, pp. 645-650.

[13]A. K. Jain and R. C. Dubes (1988). "*Algorithm for Clustering Data*", Englewood cliffs, NJ: Pretince Hall.

[14]R. Krovi (1991). *Genetic Algorithm for Clustering: A Preliminary Investigation*. IEEE Press, pp. 504-544.

[15] K. Krishna and M. Murty (1999). *Genetic K-means Algorithms*. IEEE Transactions on Systems, Man, and Cybernetics- Part-B, pp. 433-439.

[16]J. M. Adamo (2001). *Data Mining for Association Rules and Sequential Patterns*. Springer-Verlag, New York.

[17] R. Agrwal, R. Srikant (1994). *Fast Algorithms for Mining Association Rules*, in: Proceedings of the 20th International Conference on Very Large Databases, Santiago, Chile.

[18]R. Agrwal, T. Imielinski, A. Swami (1993). *Mining Association Rules Between Sets of Items in Large Databases*, in: Proceedings of ACM SIGMOD Conference on Management of Data, pp. 207-216.

[19]A. M. Ayad ((2000). *A New Algorithm for Incremental Mining of Constrained Association Rules*. Master Thesis , Department of Computer Sciences and Automatic Control, Alexandria University.

[19] C. Lance (1995). *Practical Handbook of Genetic Algorithms*. CRC Press.

[20]L. Davis (Eds.)(1991). Handbook of Genetic Algorithms, Van Nostrand , Rinhold, New York.

[21]D. E. Goldberg (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, New York.

[22]C. Z. Janikow (1993). *A Knowledge Intensive Genetic Algorithm for Supervised Learning*. Machine Learning 13, pp. 189-228.

[23] A. Giordana and F. Neri (1995). *Search Intensive Concept Induction*. Evolutionary Computation, 3(4), pp. 375-416.

[24] E. Noda, A.A. Fritas and H.S. Lopes (1999). *Discovering Interesting Prediction Rules with a Genetic Algorithm*. Proc. Conference on Evolutionary Computation (CEC-99), pp. 1322-1329. Washington D.C., USA.

[25] A.A Freitas (1998). *On Objective Measures of Rule Surprisingness*. Proc. Of 2nd European Symposium on Principle of data Mining and Knowledge Discovery (PKDD-98). Lecturer Notes in Artficial Intelligence, 1510, pp.1-9.

[26] J. M. Cover, and J. A. Thomas (1991). *Elements of Information Theory*. John Wiley & Sons.

[27]G. Syswerda (1989). *Uniform Crossover in Genetic algorithms*. Proc. Of 3rd Int. Conf. On Genetic algorithms, pp. 2-9.

[28]A. P. Engelbrecht (2002). *Computational Intelligence: An Introductions*. John Wiley & Sons.

[29]A. K. Jain, M. N. Murty, P. J. Flynn (1999). "Data Clustering: A Survey,". ACM Computing Survey, vol.31, no.3.

[30]J. C. Bezdek, S. Boggavarapu, L. O. Hall and A. Bensaid (1994). *Genetic Algorithm Guided Clustering*. In Conference Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE world Congress on Computational Intelligence, pp. 34-39.

[31]Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. Brown (2003). *Fast Genetic K-means Algorithm and its Application in Gene Expression Data Analysis*. Techincal Report TR-DB-06.

[32]J. Bhuyan (1995). *A Combination of Genetic Algorithm and Simulated Evolution Techniques for Clustering*. In C. Jinshong Hwang and Betty W. hwang, editors, proceedings of 1995 ACM Computer Science Conference, pp. 127-134.

[33]C. Alippi and R. Cucchiara (1992). *Cluster Partitioning in Image Analysis Classification: A Genetic Algorithm Approach*. In CompEuro1992 Proceedings. Computer Systems and Software Engineering, pp. 139-144.

[34]J.D. Schaffer (1984). *Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms* (Doctoral Dissertation). Nashville, TN: Vanderbilt University.

[35]J. D. Schaffer (1985). *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. Proceedings of the First International Conference on Genetic Algorithms, pp. 93-100.

[36]K. Deb (2001). *Multi-objective Optimization Using Evolutionary Algorithms*, Wiley, New York.

[37]R. E. Steuer (1986). *Multiple Criteria Optimization: Theory, Computation and Application*. Wiley, New York.

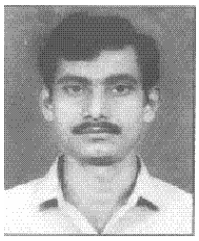
[38] R. L. Keeney and H. Raiffa (1976). *Decisions with Multiple Objectives*. John Wiley, New York.

[39]H. Eschenauer, J. Koski and A. Osyczka (1990). *Multicriteria Design Optimization*. Berlin, Springer-Verlag.

[40]A. Ghosh and S. Dehuri (2004). " *Evolutionary Algorithms for Multi-criterion optimization: A survey*", International Journal on Computing and Information Science, vol. 2, pp. 38-57.

[41]Coello Coello, Carlos A., Van Veldhuizen, David A., and Lamont, Gray B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, New York.

S. Dehuri received his M. Sc. in Mathematics from Sambalpur University in 1998, M. Tech. in Computer Science from Utkal University in 2001 and Ph.D. from Utkal University in 2006. He is currently Reader and Head at the Department of Information and Communication Technology, F. M. University, Balasore, ORISSA, INDIA.



Ashish Ghosh is a Professor of the Machine Intelligence Unit at the Indian Statistical Institute, Kolkata. He received the B.E. degree in Electronics and Telecommunication from the Jadavpur University, Calcutta in 1987, and the M.Tech. and Ph.D. degrees in Computer Science from the Indian Statistical Institute, Calcutta in 1989 and 1993, respectively.

R. Mall is a Professor of Department of Computer Science and Engineering in Indian Institute of Technology Kharagpur. He received his B. Tech, M.Tech. and Ph. D. from Indian Institute of Science, Bangalore.

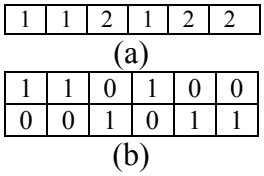


Figure 1: Chromosomes representing the clustering $\{\{O_1, O_2, O_4\}, \{O_3, O_5, O_6\}\}$ for the encoding schemes: (a) group number and (b) matrix

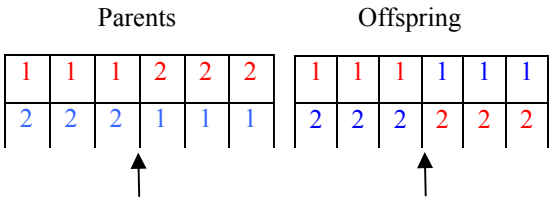


Figure 2 Context insensitivity of single–point crossover

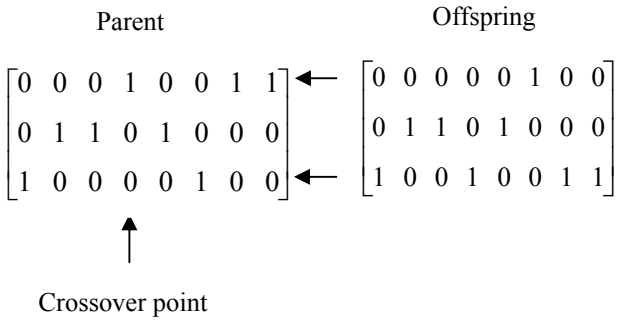


Figure 3 Alippi and Cucchiara's asexual crossover

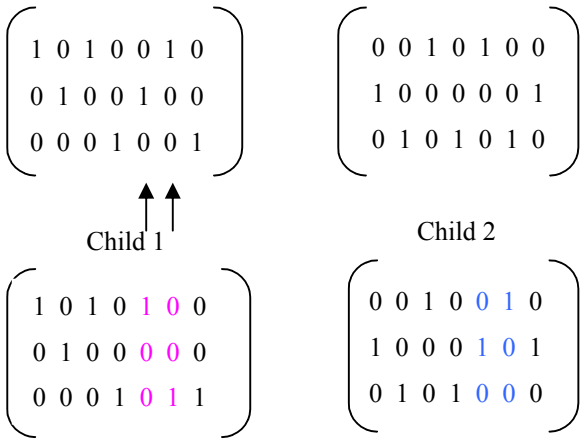


Figure 4 Bezdek et al.'s 2- point matrix crossover

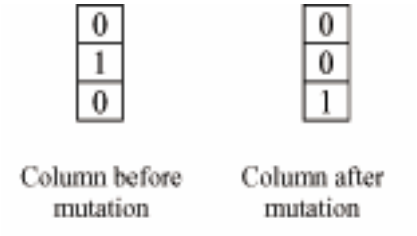


Figure 5: Column mutation

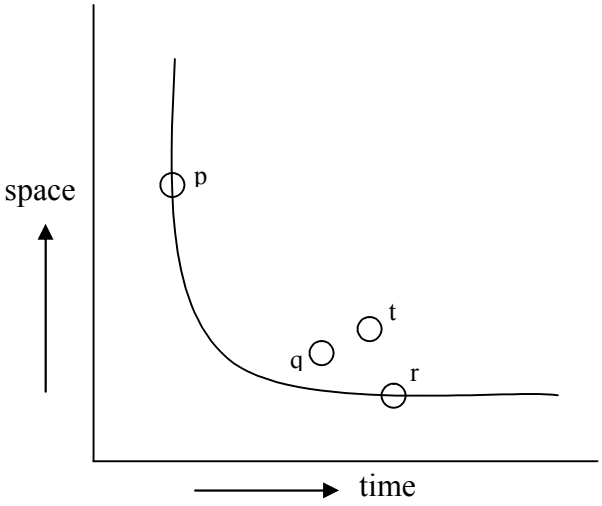


Figure 6 Trade off between time and space

Table -1

Attrib.	Values
Parents	usual, pretentious, great_pret
Has_nurs	proper, less_proper, improper, critical, very_crit
Form	complete, completed, incomplete, foster
Children	1,2,3, more
Housing	Convenient, less_conv, critical
Finance	Convenient, inconv
Social	Nonprob, slightly_prob, problematic
Health	Recommended, priority, not_recom

Table-2

Dataset	P	P_c	P_m	R_m	R_f
Zoo	100	0.8	0.03	[0, 0.7]	[0,0.8]
Nursery	500	0.75	0.002	[0.2, 0.8]	[0, 0.6]

Table 3: Rules Generated by SGA from Zoo Dataset

Class#	*Mined rules*	Fitness
1	If (hair = 1) ∧ (eggs = 0) ∧(venomous = 0) ∧(domestic = 0) Then (type =1)	0.773625
2	If (hair =0) ∧ (feathers = 1) ∧ (venomous = 0) ∧ (legs = 2) ∧ (domestic = 0) Then (type = 2)	0.765
3	If (eggs = 1) ∧ (aquatic = 0) ∧ (predator =1) (toothed=1) ∧ (fins= 0) ∧ (domestic = 0) ∧ (catsize =0) Then (type=3)	0.721
4	If (aquatic = 1) ∧ (breathes =0) ∧ (venomous =0) ∧ (tail = 1) Then (type= 4)	0.774
5	If(hair=0) ∧(airbone=0) ∧(aquatic =1) ∧(toothed = 1) ∧(breathes = 1) ∧(legs = 4) ∧(catsize=0) Then (type=5)	0.810
6	If (airbone=1) ∧(fins =0) ∧(tail =0) Then (type = 6)	0.8371
7	If (hair=0) ∧(predator=1) ∧(breathes =0) ∧(tail =0) ∧(domestic=0) Then (type=7)	0.814

Table 4: Rules Generated by MOGA from Zoo Dataset

Class#	*Mined rules*	Fitness
1	If (hair = 1) ∧ (eggs = 0) ∧(venomous = 0) ∧(domestic = 0) Then (type =1)	0.773625
2	If (hair =0) ∧ (feathers = 1) ∧ (venomous = 0) ∧ (legs = 2) ∧ (domestic = 0) Then (type = 2)	0.765
3	If (eggs = 1) ∧ (aquatic = 0) ∧ (predator =1) (toothed=1) ∧ (fins= 0) ∧ (domestic = 0) ∧ (catsize =0) Then (type=3)	0.721
4	If (aquatic = 1) ∧ (breathes =0) ∧ (venomous =0) ∧ (tail = 1) Then (type= 4)	0.774
5	If(hair=0) ∧(airbone=0) ∧(aquatic =1) ∧(toothed = 1) ∧(breathes = 1) ∧(legs = 4) ∧(catsize=0) Then (type=5)	0.810
6	If (airbone=1) ∧(fins =0) ∧(tail =0) Then (type = 6)	0.8371
7	If (hair=0) ∧(predator=1) ∧(breathes =0) ∧(tail =0) ∧(domestic=0) Then (type=7)	0.814

Table 5: Rules Generated by SGA from Nursery Dataset

Class#	Mined Rules	Predictive accuracy	Comprehensibility	Interestingness
1	If (eggs =0) ∧(venomous = 0) ∧(domestic =0) Then (type =1)	0.9090	0.8667	0.67
2	If (feathers = 1) ∧(breathes = 1) ∧(domestic =0) Then (type =2)	0.9333	0.8667	0.563
3	If (eggs =1) ∧(predator = 1) ∧(toothed =1) ∧(catsize =0) Then (type = 3)	1.0	0.8	0.563
4	If (aquatic=1) ∧(breathes=0) ∧(tail=1) Then (type =4)	0.8	0.8667	0.823
5	If(airbone=0) ∧ (aquatic =1) ∧ (toothed =1) ∧(breathes =1) ∧ (catsize=0) Then (type =5)	1.0	0.7333	0.81
6	If (airbone=1) ∧(fins=0) ∧(tail =0) Then (type =6)	0.8333	0.8	0.856
7	If (predator=1) ∧ (breathes=0) ∧ (tail=0) ∧ (domestic =0) Then (type= 7)	0.875	0.8	0.911

Table 6: Rules Generated by MOGA from Nursery Dataset

Class#	Mined Rules	Predictive accuracy	Comprehensibility	Interestingness
P	If (parents=usual) ∧(housing=less_conv) ∧(social = problematic) Then (class =P) If (parents=great_pret) ∧(social = slightly_prob) ∧(health=recommended)Then(class =P)	0.7780 0.8114	0.625	0.815

NR	If (parents=usual) ∧ (housing=less_conv) ∧ (social= slightly_prob) ∧(health=not_recom) Then (class =NR)
	If (parents=pretentious) ∧(children=3) ∧(housing= convenient) ∧(health=not_recom) Then (class =NR)
	If (parents=great_pret) ∧(children=2) ∧(housing= critical) ∧(health=not_recom) Then (class =NR)
		0.634
		0.7641
		0.783
			0.5	0.883

VR	If(housing =less_conv) ∧ (finance= inconv) ∧ (social=slightly_prob) ∧ (health = recommended) Then (class = VR)	0.897	0.5	0.761
R	If (has_nurs=proper) ∧(finance=convenient) ∧(health =recomended) Then (class =R)	0.81	0.625	0.781

Table 7: Results obtained from SGA and MOGA

Data set	Simulation	Population Size	Number of generation	Fitness of SGA	Fitness of MOGA Comp(s)	Fitness of MOGA Conn (s)
Iris	1	50	50	97.2038	97.0061	0.8
	2	50	100	98.1579	97.1079	0.6
	3	100	100	98.0944	98.0949	0.65
	4	200	100	97.1091	97.0067	0.8
	5	200	100	97.1091	97.0067	0.8
Wine	1	50	50	1.7322e4	1.7322e4	0.5
	2	50	100	1.6556e4	1.5611e4	0.67
	3	100	100	1.5611e4	1.5602e4	0.7
	4	200	100	1.5611e4	1.5611e4	0.6
	5	200	100	1.5611e4	1.5602e4	0.7
WBCD	1	50	50	3.6719e3	3.6719e3	0.65
	2	50	100	3.0485e3	3.0447e3	0.69
	3	100	100	3.0487e3	3.0487e3	0.68
	4	200	100	3.9548e3	3.3451e3	0.775
	5	200	100	3.0462e3	3.5451e3	0.775