A Distributed-Population Genetic Algorithm for Discovering Interesting Prediction Rules

Edgar Noda[1] Alex A. Freitas[2] Akebo Yamakami[1]

[1] School of Electrical and Computer Engineering (FEEC)
State University of Campinas (Unicamp), Brazil
edgar@dt.fee.unicamp.br
http://www.dt.fee.unicamp.br/~akebo

[2] Computing Laboratory
University of Kent at Canterbury
Canterbury, CT2 7NF, UK
http://www.cs.ukc.ac.uk/people/staff/aaf

Abstract

In data mining the quality of prediction rules basically involves three criteria: accuracy, comprehensible and interestingness. The majority of the rule induction literature focuses on discovering accurate, comprehensible rules. In this paper we also take these two criteria into account, but we go beyond them in the sense that we aim at discovering rules that are interesting (surprising) for the user. The search is performed by a distributed genetic algorithm (DGA) specifically designed for the discovery of interesting rules. DGAs constitute an interesting approach to tackle the premature convergence problem in evolutionary algorithms. In our approach the partition of the search space in semi-isolated subpopulations (demes) represents a subdivision of the task. We model the migration procedure of DGAs as an explicit means to promote cooperation among the demes. The algorithm addresses the dependence modeling task of data mining, where different rules can predict different goal attributes. This task can be regarded as a generalization of the very well known classification task, where all rules predict the same goal attribute. This paper also compares the results of the DGA with the results of a single population genetic algorithm to discover interesting rules.

1 Introduction

In essence, data mining consists of extracting knowledge from data [Fayyad et al. 1996]. A well-known data mining task is classification, which consists of predicting the class of an example (a record of a data set) out of a predefined set of classes, given the values of predictor attributes for that example [Hand 1997]. This paper addresses a kind of generalization of the classification task, called dependence modeling [Freitas 2000], where there are several goal attributes to be predicted, rather than just one goal attribute. In this context, we address the discovery of prediction rules of the form:

IF some conditions on the values of predictor attributes are true
THEN predict a value for some goal attribute.

In our approach for dependence modeling the user specifies a small set of potential goal attributes, which she/he is interested in predicting. Although we allow more than one goal attribute, each prediction rule has a single goal attribute in its consequent (THEN part). However, different rules can have different goal attributes in their consequent.

In principle, the prediction rules discovered by a data mining algorithm should satisfy three properties, namely: predictive accuracy, comprehensibility and interestingness [Freitas 2002]. In this paper we propose a distributed-population genetic algorithm (GA) designed to discover a few rules that are both interesting and accurate. Both these criteria are included in the fitness function of the GA. In addition, designating, as the output of the GA, a small set of rules, which can be thought of as "knowledge nuggets" extracted from the data, facilitates the discovery of comprehensible knowledge.

Discovered knowledge should also be comprehensible to the user. Assuming that the output of the data mining algorithm will be used to support a decision ultimately made by a human being, knowledge comprehensibility is an important requirement [Spiegelhalter et al. 1994]. Knowledge represented as high-level rules, as in the above-mentioned IF-THEN format, has the advantage of being closely related to natural language. Therefore, the output of rule discovery algorithms tends to be more comprehensible than the output of other kinds of algorithms, such as neural networks and various statistical algorithms.

Discovered knowledge should also be interesting to the user. Among the three above-mentioned desirable properties of discovered knowledge, interestingness seems to be the most difficult one to be quantified and to be achieved. By “interesting” we mean that discovered knowledge should be novel or surprising to the user. We emphasize that the notion of interestingness goes beyond the notions of predictive accuracy and comprehensibility. Discovered knowledge may be highly accurate and comprehensible, but it is uninteresting if it states the obvious or some pattern that was previously known by the user. A very simple, classical example shows the point. Suppose one has a medical database containing data about a hospital’s patients. A data mining algorithm could discover the following rule from such a database: IF (patient is pregnant) THEN (patient is female). This rule has a very high predictive accuracy and it is very comprehensible. However, it is uninteresting, since it states an obvious, previously known pattern.

2 GA-Nuggets

In our previous work we have introduced a GA for dependence modeling, called GA-Nuggets [Noda et al. 1999]. This GA maintains a single, centralized population of individuals. In this paper we propose a major extension of that GA. It maintains a distributed population, consisting of several subpopulations, each of them evolving in an independent manner, with occasional migration between them. Subsection 2.1 briefly reviews the main aspects of GA-Nuggets (see [Noda et al. 1999] for details), whereas the new distributed-population scheme is described in subsection 2.2. For am overview of distributed GAs in general the reader is referred to [Herrera et al. 1999] and [Cantú-Paz 2000].

2.1 Single-Population GA-Nuggets

Each individual represents a candidate prediction rule of the form: IF *Ant* THEN *Cons*, where *Ant* is the rule antecedent and *Cons* is the rule consequent. *Ant* consists of a conjunction of conditions, where each condition is an attribute-value pair 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 domain of A_i. An individual is encoded as a fixed-length string containing *z* genes (see figure 1), where *z* is the number of attributes (considering both predictor and goal attributes). Only a subset of the attribute values encoded in the genome will be decoded into attribute values occurring in the rule antecedent. Therefore, although the genome length is fixed, its decoding mechanism effectively represents a variable-length rule antecedent.

A_1	A_2	A_3	. . .	. . .	. . .	A_Z

Figure 1: Individual representation

Once the rule antecedent is formed, the algorithm chooses the best consequent for each rule in such a way that maximizes the fitness of an individual (candidate rule). In effect, this approach gives the algorithm some knowledge of the data-mining task being solved. This approach can also be seen as an efficient way of implementing a genetic search for rules. For a given rule antecedent, with a single scan of the training set the system is actually evaluating several different candidate rules and choosing the best one. Therefore, the bottleneck of fitness evaluation – viz., scanning the training set – is performed just once in order to evaluate multiple candidate rules.

The fitness function consists of two parts. The first one measures the degree of interestingness of the rule, while the second measures its predictive accuracy. 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 (*AntInt*) is calculated by an information-theoretical measure [Freitas 1998]. In formula [1], *n* is the number of attributes occurring in the rule antecedent and $|Dom(G_k)|$ is the domain cardinality (i.e. the number of possible values) of the goal attribute G_k occurring in the consequent. The log term is included in formula [1] to normalize the value of *AntInt*, so that this measure takes on a value between *0* and *1*.

The computation of the rule consequent’s degree of interestingness (*ConsInt*) is based on the idea that the prediction of a rare goal attribute value tends to be more interesting to the user than the prediction of a very common goal attribute value [Freitas 1999]. In formula [2] $Pr\ (G_{kl})$ is the prior probability (relative frequency) of the goal attribute value G_{kl}, and β is a user-specified parameter, empirically set to *2* in our experiments. The exponent $1/\beta$ in the equation [2] can be regarded as a way of reducing the influence of the rule consequent interestingness in the value of the fitness function.

The computation of these two degrees of interestingness is described in detail in [Noda et al. 1999]. The second part of the fitness function measures the predictive accuracy (*PredAcc* – formula

[3]) of the rule. Where |*A&C*| is the number of examples that satisfy both the rule antecedent and the rule consequent, and |*A*| is the number of examples that satisfy only the rule antecedent. The term ½ is subtracted in the numerator of formula [6] to penalize rules covering few training examples – see [Quinlan 1987].

Formula [4] is the final fitness function. Where W_1 and W_2 are user-defined weights. In our experiment they are set to 1 and 2, respectively.

$$AntInt = 1 - \left(\frac{\frac{\sum_{i=1}^{n} InfoGain(A_i)}{n}}{\log_2(|\, dom(G_k)\,|)} \right) \qquad [1]$$

$$ConsInt = (1 - \Pr(G_{kl}))^{1/\beta} \qquad [2]$$

$$PredAcc = \frac{|\, A \,\&\, C\,| - 1/2}{|\, A\,|} \qquad [3]$$

$$Fitness = \frac{w_1 . \frac{AntInt + ConsInt}{2} + w_2 . PredAcc}{w1 + w2} \qquad [4]$$

GA-Nuggets uses a well-known tournament selection method with tournament size 2. The algorithm uses uniform crossover extended with a "repair" procedure. After the standard crossover is done, the algorithm checks if any invalid individual was created. If so, a repair procedure is performed to produce valid-genotype individuals. The mutation operator randomly transforms the value of an attribute into another value belonging to the domain of that attribute.

There are two operators, called condition-insertion and condition-removal operators, which control the size of the rules being evolved by randomly inserting/removing a condition into/from a rule antecedent. The probability of applying each of these operators depends on the current number of conditions in the rule antecedent. The larger the number of conditions in the current rules antecedent, the smaller the probability of applying the condition-insertion operator.

2.2 Distributed-Population GA-Nuggets

In this new version of GA-Nuggets, the entire population is divided into *p* subpopulations, where *p* is the number of goal attributes. In each subpopulation all individuals are associated with the same goal attribute. The individual representation of the distributed-population version of GA-Nuggets is similar to the individual representation of the single-population version of GA-Nuggets, described in subsection 2.1. The only difference is that the goal attribute is fixed for all individuals of the same subpopulation. Each subpopulation evolves independently from the others (except for some occasional migrations).

One advantage of this distributed population approach, with a fixed goal attribute for each subpopulation, is to reduce the number of crossovers performed between individuals predicting different goal attributes. Since crossover is restricted to individuals of the same subpopulation, crossover swaps genetic material of two parents, which represent candidate rules predicting the same goal attribute. Note that this is not the case with single-population GA-Nuggets, where crossover can swap genetic material between parents representing rules predicting different goal attributes.

Distributed-population GA-Nuggets has a migration procedure [Cantú-Paz 2001] where, from time to time, an individual of a subpopulation is copied into another subpopulation. We have developed a migration procedure tailored for our prediction-rule discovery task, as follows. The subpopulations evolve in a synchronous manner, so that in each subpopulation the *i*-th generation is started only after the (i - 1)-th generation has been completed in all subpopulations, for $i = 2, ..., g$, where g is the number of generations (which is the same for all subpopulations).

Migration takes place every m generations. Each subpopulation sends individuals to all the other subpopulations. More precisely, in each subpopulation S_i, $i = 1,..., p$, the migration procedure chooses (p - 1) individuals to be migrated. Each of those p -1 migrating individuals will be sent to a distinct subpopulation S_j, $j = 1,..., p$, $j \neq i$. The choice of the individuals to be migrated is driven by the fitness function, taking into account the fact that different subpopulations are associated with different goal attributes. In each subpopulation S_i the migration procedure knows, for each individual, not only the actual value of its fitness in that subpopulation (called its *home* fitness), but also what would be the value of the fitness of that individual if it were placed in another subpopulation S_j, $j = 1,..., p, j \neq i$, predicting a value of the j-th goal attribute. The latter is called the *foreign* fitness of the individual in subpopulation S_j. Subpopulation S_i sends to subpopulation S_j a copy of the individual with maximum foreign fitness in S_j.

On the other hand, each subpopulation S_i, $i = 1,..., p$, receives $p - 1$ individuals, each coming from a different subpopulation S_j, $j = 1,..., p, j \neq i$. Among these incoming $p - 1$ individuals, only one is accepted by subpopulation S_i. The accepted individual is the one with the largest fitness value. This is equivalent to a tournament selection among the incoming individuals.

The fitness function of distributed-population GA-Nuggets is the same as the fitness function of single-population GA-Nuggets, as defined by formula [1]. In distributed-population GA-Nuggets the application of the selection method and genetic operators is independently performed in each of the subpopulations. Each subpopulation uses the same selection method and genetic operators (described in subsection 2.1.3), which are applied only to the local individuals in that subpopulation.

3 Computational Results and Discussion

The data sets used to evaluate the previously described algorithms were obtained from the UCI repository of machine learning databases (*http://www.ics.uci.edu/AI/Machine-Learning.html*). The data sets used are Zoo, Car Evaluation, Auto Imports and Nursery. They are normally used for evaluating algorithms performing the classification task. In the absence of a specific benchmark data set for the dependence-modeling task, these data sets were chosen because they seem to contain more than one potential goal attribute.

The zoo database 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, since this attribute has no generalization power. The attributes in the zoo data set are all categorical. The attribute names are as follows: hair, feathers, eggs, milk, predator, toothed, domestic, backbone, fins, legs, tail, catsize, airborne, aquatic, breathes, venomous and type. Except type and legs, the attributes are Boolean. In our experiments the set of potential goal attributes used was predator, domestic and type. Predator and domestic are Boolean attributes, whereas the type attribute can take on seven different values. The car evaluation dataset contains 1728 instances and 6 attributes. All attributes are categorical and there are no missing values. The attributes names are buying, maint, doors, persons, lug_boot, safety and car acceptability. The attributes buying and car acceptability were used as potential goal attributes. The auto-imports 85M dataset contains 205 instances and 26 categorical attributes. The attribute normalized-losses and 12 instances were removed because of missing values. The attributes symboling, body-style and price, with 7, 5, and 3 values, were chosen as goals. The nursery school data set contains 12960 instances and 9 attributes. The attributes are all categorical. The attribute names are as follows: parents, health, form, children, finance, housing, social, has_nurs and recommendation. In our experiments, the attributes used as potential goal attributes were finance, social and health.

We emphasize that in our approach for dependence modeling we do not aim at classifying the whole test set. Rather, the goal is to discover a few interesting rules to be shown to a user. We can think of the discovered rules as the most valuable “knowledge nuggets” extracted from the data. These knowledge nuggets are valuable even if they do not cover the whole test set. In other words, the value of the discovered rules depends on their predictive accuracy on the part of the test set covered by those rules, but not on the test set as a whole. For each data set we have run a 10-fold cross-validation procedure [Hand 1997] to evaluate the quality of the rules discovered by two algorithms, namely: single-population GA-Nuggets (section 2.1), and distributed-population GA-Nuggets (section 2.2). The computational experiments measured both the predictive accuracy (accuracy rate in the test set) and the degree of interestingness of the rules discovered by the two algorithms.

3.1 Predicative accuracy

In this subsection, we compare the results, for all datasets, of the two versions of the GA, concerning the predictive accuracy issue. In Tables 1, 2, 3 and 4 the first column is the name of the goal attribute, followed by its possible values. Each row of these tables corresponds to a discovered rule whose consequent (THEN part) is defined by the combination of the goal attribute and value specified in the first two columns. The next columns contain, for each version of the GA, the coverage (number of examples satisfying the rule antecedent) and the predictive accuracy of the corresponding rule on the test set. The numbers after the "±" symbol denote standard deviations.

Tables 1, 2, 3 and 4 show the results for the Zoo, Car Evaluation, Auto Imports and Nursery data sets, respectively. Considering the single-population GA as a baseline, in the last column of these tables the sign (+) indicates that the distributed-population GA significantly outperformed the baseline, whereas the sign (-) indicates the opposite, i.e., the baseline significantly outperformed the distributed-population GA. The difference of predictive accuracy between the two versions of the GA was considered significant when the corresponding one-standard deviation intervals do not overlap each other.

With respect to predictive accuracy distributed-population GA-Nuggets obtained somewhat better results than single-population GA-Nuggets. In only one case (in Table 4) the single-population GA-Nuggets found rules with significantly higher predictive accuracy. Distributed GA-Nuggets significantly outperformed single-population GA in six cases (one case in Table 1, three cases in Table 3, and two cases in Table 4).

Table 1: Predictive Accuracy (%) in the Zoo data set

Goal Attrib.	Attrib. Value	GA		Distrib. GA	
		Cov.	Pred. Acc.	Cov.	Pred. Acc.
Predator	False	4.4	50.5 ± 8.9	3.2	48.0 ± 8.2
	True	2.8	75.0 ± 11.2	2.4	84.0 ± 11.1
Domestic	False	5.2	97.1 ± 5.2	6.2	90.5 ± 4.4
	True	0.8	0.0 ± 0.0	0.8	0.0 ± 0.0
Type	1	6.4	100.0 ± 0.0	6.4	100.0 ± 0.0
	2	3.6	100.0 ± 0.0	3.6	100.0 ± 0.0
	3	0.2	0.0 ± 0.0	1.1	95.0 ± 13.8 (+)
	4	2.2	100.0 ± 0.0	2.2	100.0 ± 0.0
	5	0.5	100.0 ± 0.0	0.8	100.0 ± 0.0
	6	1.1	90.0 ± 10.0	1.1	90.0 ± 10.0
	7	2.0	83.3 ± 10.2	2.0	85.0 ± 11.0

Table 2: Predictive Accuracy (%) in the Car Evaluation data set

Goal Attrib.	Attrib. Value	GA		Distrib. GA	
		Cov.	Pred. Acc.	Cov.	Pred. Acc.
Buying	V-high	1.2	60.0 ± 16.3	1.0	50.0 ± 16.7
	High	2.5	4.5 ± 3.0	2.2	7.5 ± 3.8
	Med	2.5	2.5 ± 2.5	2.3	5.0 ± 3.3
	Low	2.3	100.0 ± 0.0	2.0	100.0 ± 0.0
Accept.	Unacc	10.4	100.0 ± 0.0	10.4	100.0 ± 0.0
	Acc	0.1	0.0 ± 0.0	0.0	0.0 ± 0.0
	Good	0.0	0.0 ± 0.0	0.1	0.0 ± 0.0
	V-good	0.0	0.0 ± 0.0	0.1	0.0 ± 0.0

3.2 Degree of Interestingness

The computational results with respect to the degree of interestingness of the discovered rules are reported in Tables 5, 6, 7 and 8, whose structure is similar to the structure of Tables 1, 2, 3, and 4. The main difference is that, instead of coverage and predictive accuracy results, Tables 5 to 8 contains columns reporting the degree of interestingness of the rule consequent (Cons. Int.) and the degree of interestingness of the rule antecedent, both expressed in %. Tables 5, 6, 7, and 8 report results for the Zoo, Car Evaluation, Auto Imports and Nursery data sets, respectively. Again, the single-population GA was considered as a baseline, and in the last column of these tables the sign (+) indicates that the distributed-population GA significantly outperformed the baseline, whereas the sign (-) indicates the opposite.

With respect to the degree of interestingness of the discovered rules, distributed-population GA-Nuggets obtained results considerably better than single-population GA-Nuggets. More precisely, the former significantly outperformed the latter in 22 out of 44 cases – considering all the discovered rules in all the four data sets – whereas the reverse was true in just five out of 44 cases. In the other cases the difference between the two algorithms was not statistically significant.

Table 3: Predictive Accuracy (%) in the Auto Imports data set

Goal Attrib.	Attrib. Value	GA		Distrib. GA	
		Cov.	Pred. Acc.	Cov.	Pred. Acc.
Simb.	-3	0.0	0.0 ± 0.0	0.0	0.0 ± 0.0
	-2	0.0	0.0 ± 0.0	0.8	0.0 ± 0.0
	-1	1.2	55.0 ± 13.8	1.6	63.3 ± 14.4
	0	2.2	96.0 ± 2.7	2.0	98.0 ± 2.0
	1	1.7	70.0 ± 15.3	2.3	70.0 ± 10.2
	2	1.2	63.3 ± 14.4	1.3	90.0 ± 10.0 (+)
	3	1.2	70.0 ± 15.3	1.9	70.0 ± 12.6
Body	Hardtop	0.6	0.0 ± 0.0	0.4	0.0 ± 0.0
	Wagon	0.6	0.0 ± 0.0	1.6	13.3 ± 5.4 (+)
	Sedan	0.6	60.0 ± 16.3	2.1	82.5 ± 9.9 (+)
	Hatch	2.6	76.7 ± 6.7	2.8	71.7 ± 5.4
	Convert.	0.6	40.0 ± 16.3	1.0	25.0 ± 8.3
Price	Low	11.4	100.0 ± 0.0	13.4	100.0 ± 0.0
	Average	3.2	90.0 ± 4.1	3.7	81.7 ± 9.7
	High	1.4	72.5 ± 12.6	1.3	90.0 ± 10.0

Table 4: Predictive Accuracy (%) in the Nursery data set

Goal Attrib.	Attrib. Value	GA		Distrib. GA	
		Cov.	Pred. Acc.	Cov.	Pred. Acc.
Finance	Conv.	2.2	80.0 ± 13.3	3.4	100.0 ± 0.0(+)
	Inconv.	3.4	100.0 ± 0.0	3.9	100.0 ± 0.0
Social	Non-prob	3.2	1.11 ± 1.1	2.2	0.0 ± 0.0
	Slightly prob	27.7	6.4 ± 4.3	2.0	0.0 ± 0.0 (-)
	Problem.	4.4	100.0 ± 0.0	10.2	100.0 ± 0.0
Health	Recomm.	0.0	0.0 ± 0.0	0.0	0.0 ± 0.0
	Priority	291.6	0.0 ± 0.0	0.2	0.0 ± 0.0
	Not recomm.	54.5	12.8 ± 9.8	15.8	41.8 ± 14.4(+)
	Spec priority	4.6	100.0 ± 0.0	10.0	100.0 ± 0.0
	Very recomm.	10.8	100.0 ± 0.0	8.6	100.0 ± 0.0

Table 5: Interestingness (%) in Zoo data set

Goal Attrib.	Attrib. Value	Cons. Int.	Antecedent Interestingness	
			GA	Distrib. GA
Predator	False	74.4	97.5 ± 0.4	95.9 ± 1.0 (-)
	True	66.8	94.9 ± 0.5	96.4 ± 0.4 (+)
Domestic	False	35.7	96.3 ± 0.5	96.9 ± 0.6
	True	93.3	96.9 ± 0.7	97.9 ± 0.4
Type	1	77.1	94.7± 0.2	94.6 ± 0.1
	2	89.0	93.9 ± 0.3	93.9 ± 0.3
	3	97.5	93.2 ± 0.6	92.3 ± 0.2 (-)
	4	94.3	93.4 ± 0.2	94.7 ± 0.3 (+)
	5	97.9	94.3 ± 0.4	94.0 ± 0.3
	6	95.9	93.4 ± 0.3	92.4 ± 0.4 (-)
	7	94.9	95.3 ± 0.1	95.1 ± 0.2

We have also observed that distributed-population GA-Nuggets has performed a more cost-effective search than single-population GA-Nuggets, in the sense that in general the former has obtained good solutions in earlier generations, by comparison with the latter. (Both versions of the GA had the same total population size, so that the comparison was fair.) Hence, overall, considering the results in the four data sets, distributed-population GA-Nuggets represents an improvement over single-population GA-Nuggets.

Table 6: Interestingness (%) in Car Evaluation data set

Goal Attrib.	Attrib. Value	Cons. Int.	Antecedent Interestingness	
			GA	Distrib. GA
Buying	V-high	86.6	99.4 ± 0.0	99.4 ± 0.0
	High	86.6	99.4 ± 0.0	99.4 ± 0.0
	Med	86.6	99.3 ± 0.0	99.4 ± 0.0 (+)
	Low	86.6	98.8 ± 0.0	99.0 ± 0.0(+)
Accept.	Unacc	54.7	96.5 ± 0.0	96.4 ± 0.0 (-)
	Acc	88.3	93.2 ± 0.0	93.3 ± 0.0 (+)
	Good	97.9	94.3 ± 0.0	94.3 ± 0.0
	V-good	98.1	94.3 ± 0.0	94.3 ± 0.0

Table 7: Interestingness (%) in Auto Imports data set

Goal Attrib.	Attrib. Value	Cons. Int.	Antecedent Interestingness	
			GA	Distrib. GA
Simb.	-3	100.0	99.3 ± 0.1	100.0 ± 0.0 (+)
	-2	99.2	98.3 ± 0.1	99.0 ± 0.3 (+)
	-1	94.1	97.7 ± 0.1	97.8 ± 0.1 (+)
	0	82.1	97.7 ± 0.2	97.5 ± 0.1
	1	85.8	97.8 ± 0.2	97.9 ± 0.1
	2	91.6	97.4 ± 0.2	98.1 ± 0.1 (+)
	3	93.8	98.1 ± 0.1	98.7 ± 0.1(+)
Body	Hardtop	97.9	97.5 ± 0.3	98.3 ± 0.4 (+)
	Wagon	93.6	97.6 ± 0.2	98.1 ± 0.3
	Sedan	72.3	96.5 ± 0.5	97.8 ± 0.5 (+)
	Hatch	82.1	97.1 ± 0.3	97.5 ± 0.1
	Convert.	98.4	98.1 ± 0.2	98.6 ± 0.1 (+)
Price	Low	64.8	94.2 ± 0.5	96.8 ± 0.1 (+)
	Average	80.8	92.9 ± 0.9	95.1 ± 0.3 (+)
	High	96.3	90.8 ± 0.4	96.1 ± 0.2 (+)

Table 8: Interestingness (%) in Nursery data set

Goal Attrib.	Attrib. Value	Cons. Int.	Antecedent Interestingness	
			GA	Distrib. GA
Finance	Conv.	71.1	99.8 ± 0.0	99.9 ± 0.0 (+)
	Inconv.	70.3	99.8 ± 0.0	99.9 ± 0.0 (+)
Social	Non-prob	81.7	99.7 ± 0.0	99.9 ± 0.0 (+)
	Slightly prob	81.6	99.8 ± 0.0	99.9 ± 0.0 (+)
	Problem.	81.6	99.7 ± 0.0	99.8 ± 0.0 (+)
Health	Recomm.	81.7	94.9 ± 0.0	94.9 ± 0.0
	Priority	99.9	99.7 ± 0.0	99.9 ± 0.0 (+)
	Not recomm.	98.7	96.3 ± 0.7	94.6 ± 0.4 (-)
	Spec priority	81.9	93.5 ± 0.3	93.4 ± 0.3
	Very recomm.	82.9	94.1 ± 0.3	94.3 ± 0.3

4 Conclusion and future works

In this paper we have presented two algorithms for discovering "knowledge nuggets" – rules that have both a good predictive accuracy and a good degree of interestingness. The algorithms were developed for discovering prediction rules in the dependence modeling task of data mining. This task can be regarded as a generalization of the very well known classification task.

The algorithms presented in this paper are actually two different versions of a Genetic Algorithm (GA). One of these versions uses a single population of individuals, whereas the other version uses a distributed population of individuals. With the exception of this major difference, the other characteristics of the GA were kept the same, as much as possible, in the two versions, in order to allow us to compare the two versions in a manner as fair as possible.

This comparison was performed across four public domain, real-world data sets. The computational experiments measured both the predictive accuracy (accuracy rate in the test set) and the degree of Interestingness of the rules discovered by the two algorithms.

As discussed in section 3, overall the computational results indicate a somewhat better performance of the distributed approach, with respect to predictive accuracy. With respect to the degree of interestingness of the discovered rules, the distributed-population version of the GA obtained results considerably better than the single population algorithm.

One direction for future research consists of developing a new version of the distributed-population GA where each subpopulation is associated with a goal attribute value, rather than with a goal attribute as in the current distributed version. It will be interesting to compare the performance of this future version with the performance of the current distributed version, in order to empirically determine the cost-effectiveness of these approaches. It would also be useful to extend the computational experiments reported in this paper to other data sets, and other migration policies to further validate the reported results.

References

[Cantú-Paz 2000] E. Cantú-Paz. *Efficient and Accurate Parallel Genetic Algorithms.* Kluwer Academic Publisher, 2000.

[[Cantú-Paz 2001] E. Cantú-Paz. Migration Policies, Selection Pressure, and Parallel Evolutionary Algorithms. *Journal of Heuristics*, 7 (4), 311-334, 2001.

[Fayyad et al. 1996] U.M. Fayyad, G. Piatetsky-Shapiro and P. Smyth. 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 and Data Mining*, 1-34. AAAI/MIT Press, 1996.

[Freitas 1998] A.A. Freitas. On objective measures of rule surprisingness. *Principles of Data Mining and Knowledge Discovery (Proceedings of the 2nd European Symp., PKDD´98) – Lecture Notes in Artificial Intelligence 1510,* 1-9. Springer, 1998.

[Freitas 1999] A.A. Freitas. A genetic algorithm for generalized rule induction. In: R. Roy et al. *Advances in Soft Computing - Engineering Design and Manufacturing*. (*Proceedings of the WSC3, 3rd on-line world conf., hosted on the internet, 1998)*, 340-353. Springer, 1999.

[Freitas 2000] A.A. Freitas. Understanding the crucial differences between classification and discovery of association rules - a position paper. ACM SIGKDD Explorations, 2(1), 65-69. ACM, 2000.

[Freitas 2002] A.A. Freitas. *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. (Forthcoming book.) Berlin: Springer-Verlag, 2002.

[Hand 1997] D.J. Hand. *Construction and Assessment of Classification Rules*. John Wiley&Sons, 1997.

[Herrera et al. 1999] F. Herrera et al. Hierarchical Distributed Genetic Algorithms. *International Journal of Intelligent Systems*, 14(11), 1099-1121, November 1999.

[Noda et al. 1999] E. Noda, A.A. Freitas, and H.S. Lopes. Discovering interesting prediction rules with a genetic algorithm. *Proc. of the Congress on Evolutionary Computation (CEC-99)*, pp. 1322-1329. IEEE Press, 1999.

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

[Quinlan 1987] J.R. Quinlan. Generating production rules from decision trees. *Proc. of the Tenth Int. Joint Conf. on Artificial Intelligence (IJCAI-87)*, 304-307. San Francisco: Morgan Kaufmann, 1987.