An Evolutionary Algorithm for Automated Discovery of Small-Disjunct Rules

Basheer M. Al-Maqaleh*
Faculty of Computer Sciences & Information Systems, Thamar University, Yemen

Mohammed A. Al-Dohbai
Faculty of Computer Sciences & Information Systems, Thamar University, Yemen

Hamid Shahbazkia
Faculty of Science and Technology, University of Algarve, Faro, Portugal

ABSTRACT
In general rule induction algorithms have a bias that favors the discovery of large disjuncts, rather than small disjuncts. In the context of data mining, small disjuncts are rules covering a small number of examples. Due to their nature, small disjuncts are error prone. It correctly classify individually only few examples but, collectively, cover a significant percentage of the set of examples, so that it is important to develop new approaches to cope with the problem of small disjuncts. This paper presents a classification algorithm based on Evolutionary Algorithm (EA) that discovers interesting small-disjunct rules in the form **If** P **Then** D. The proposed system specifically designed for discovering rules covering examples belonging to small disjuncts. The proposed algorithm is validated on several datasets of UCI data set repository and the experimental results are presented to demonstrate the effectiveness of the proposed scheme for automated small-disjunct rules mining.

General Terms
Knowledge Discovery in Databases (KDD), Data Mining, Evolutionary Algorithms.

Keywords
Interestingness, Small disjunct, Predictive accuracy, Genetic algorithm.

1. INTRODUCTION
Knowledge Discovery in Databases (KDD) is defined as the nontrivial process of identifying valid, novel, potentially useful and ultimately understandable patterns of data [1]. Data mining is a core stage in the entire process of KDD which applies an algorithm to extract patterns of data[2]. The idea of automatically discovering knowledge from databases is a very attractive and challenging task, both for academia and for industry. Hence, there has been a growing interest in data mining in several machine learning related areas, including Evolutionary Algorithms (EAs). The paradigm of EAs consists of stochastic search algorithms that are based upon Darwin's natural selection theory of evolution, where a population is progressively improved by selectively discarding the worse and breeding new offspring from the better[3]. EAs are often used as optimization algorithms, and this is the role that they play in most data mining applications. The main motivation for applying EAs to KDD tasks is that they are robust and adaptive search methods, which perform a global search in the space of candidate solutions. Intuitively, the global search performed by EAs can more effectively discover interesting patterns that would have been missed by the greedy search performed by many KDD methods[4]. Classification in data mining is a technique that is used to predict group membership of data examples in a dataset. An example or a record consists of a set of predicting attributes and a goal attribute. General relationships between predicting and goal attributes are discovered on training data and then these relationships are validated on test data[2],[5]. Classification rule mining is one of the most studied tasks in data mining community because the data being generated and stored in databases of organizations are already enormous and continue to grow very fast. This large amount of stored data normally contains valuable hidden knowledge, which if discovered, could be used to improve the decision-making process of an organization [6]. Among the several kinds of knowledge representation that can be used to represent the knowledge discovered by a classification algorithm is **If-Then** classification rules of the form: **If** <some conditions are satisfied> **Then** <predict the value of some goal attribute>. We aim at discovering rules whose consequent (**Then** part) predict the value of some goal attribute for an example that satisfies all the conditions in the antecedent (**If** part) of the rule. This knowledge representation has the advantage of being intuitively comprehensible for the user[4]. A small disjunct can be defined as a rule which covers a small number of training examples. Also, a small disjunct covers a small number of examples, the set of all small disjuncts can cover a large number of examples[7]. For instance, Danyluk and Provost [8] report a real-world application where small disjuncts cover roughly 50% of the training examples. Large disjuncts discovery is performed by using general rule induction. The small disjuncts are therefore left behind by using general rules as normally capturing generalization is preferred to capturing specialization over a training set, since the latter are unlikely to be valid in the test set. In this way small disjuncts are not considered in discovering the rule set, since they tend to be error prone. Nevertheless a closer look shows that small disjuncts are interesting in the context of data mining, for the following reasons[9],[10]:

(a) Data mining has also a goal of discovering previously-unknown rules. General rules are normally considered to be known but small-disjunct rules are harder to discover. Therefore, including small disjunct allow these unknown rules to be discovered.

(b) It is interesting to discover small disjunct, as they dispute the existing knowledge and have elements of unexpectedness and interestingness. Small disjuncts add valuable evidences that may be few in number but enhance the quality of decision making towards perfection. Knowing small-disjunct rule helps

* Basheer Mohamad Al-Maqaleh –Corresponding author- is currently doing his Post-Doctoral Research at Faculty of Science and Technology, University of Algarve, Faro, Portugal. (e-mail: basheer.almaqaleh.dm@gmail.com).

us to make right decisions even in rare and exceptional circumstances. As small disjunct focus on a very small portion of data, discovering small-disjunct rule still remains a great challenge. Genetic Algorithms(GAs) are robust, flexible algorithms, which tend to cope well with attribute interactions. Hence, they can be more easily tailored for coping with small disjuncts, which are associated with large degrees of attribute interaction[11],[12]. The purpose of this paper is to discover comprehensible, and interesting small-disjunct rules as underlying knowledge representation in the form of classification rule sets from datasets using GA. The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 presents new GA design that includes encoding scheme, genetic operators applied and fitness function. Section 4 reports the results of experiments evaluating the performance of our system on datasets used. Conclusions and future directions are given in Section 5.

2. RELATED WORK

A Numerous attempts have been made to apply EAs in data mining to tackle the problem of knowledge extraction and classification. Several GA designs, for discovering comprehensible classification rules, have been proposed in the literature[6],[13],[14],[15],[16]. A new technique for organizing discovered rules in different levels of detail is introduced by Liu et al. [17]. They call these rules the top-level general rules. The second is to find exceptions, exceptions of the exceptions and so on. Some of the exception rules found by this method could be considered as small disjuncts. Weiss and Hirsh present a quantitative measure for evaluating the effect of small disjuncts on learning [18]. Their results confirmed that small disjuncts do have a negative impact on predictive accuracy in many cases. Gomes presents a new algorithm to discover small disjuncts [19]. In his paper, covering and accuracy measures are used to identify small disjuncts groups. Weiss [20] suggests that there is a relation between the problem of small disjuncts and class imbalance, stating that one of the reasons why small disjuncts have a higher error rate than large disjuncts is due to class imbalance. Holte et al. [21] and Weis [22] show that small disjuncts often correspond to rare cases within the domain under study and cannot be totally eliminated if high predictive accuracy is to be achieved. They argue with justification that learned concepts must be able to include small disjuncts arising from exceptions and rare cases. Their work offers an approach to reducing the risk of using small disjuncts by changing bias. The use of a hybrid data mining methods to cope with small disjuncts are proposed by[12]. The basic idea is to use a well-known decision-tree algorithm to classify examples belonging to large disjuncts and use a genetic-algorithm to discover rules classifying examples belonging to small disjuncts. Decision tree and rule inducing methods implement a maximum generality bias; wherever they create a new rule, it is made as general as possible while still discriminating the concept being learned. This approach favors more general rules or large disjuncts. Ting proposed the use of a hybrid data mining algorithm to cope with small disjuncts[23]. His method consists of using a decision-tree algorithm to cope with large disjuncts and an instance-based learning algorithm to cope with small disjuncts. In the present work, the proposed algorithm is designed to discover only small-disjunct rules from datasets.

3. THE PROPOSED GA SYSTEM

GAs are stochastic search methods which imitate the natural phenomena of survival of the fittest and genetic inheritance. In this section, we describe our GA developed for discovering small-disjunct rules.

3.1 Encoding

Our encoding follows Michigan approach where each individual represents a small-disjunct rule. A small-disjunct rule is represented as: **If** P **Then** D; where P is premise which is a conjunction of terms. Each term is an attribute value pair. **If** part of the rule consists of a conjunction conditions on the values of at most *n*-1 predicting attributes, where *n* is the number of attributes being mined .Decision D is a single term that contains the goal attribute. All the predicting and goal

Table 1. Description of the object dataset

Attribute	Possible values	Allies
size	medium, small, large	'1', '2','3'
color	blue, red green	'1', '2','3'
shape	brick, wedge, sphere, pillar	'1', '2','3','4'
decision	no, yes	'1', '2'

attribute values are categorical and coded from '1' to '9' and if need be from 'a' to 'z' [24],[25]. For example, consider the small object classification dataset[26] given in Table 1.

Corresponding to the above dataset, a small-disjunct rule: **If** size =medium ∧ color=blue **Then** decision=yes, would be encoded as:

Size	color	shape	decision
Gene1	Gene 2	Gene 3	Gene 4
1	1	#	2

The genes are positional, i.e. the first gene represents the first attribute, the second gene represents the second attribute and so on. If an attribute is not present in the rule antecedent, the corresponding value in gene is ‘‘#’’. This value is a flag to indicate that the attribute does not occur in the rule antecedent. Though the above encoding is fixed length, it is flexible and the genes are interpreted in such a way that individual phenotypes in the GA population have variable length individuals having different number of conditions. This kind of representation gives a lot of flexibility to the rules being discovered[4],[14].

3.2 Genetic Operators

Genetic operators are one of the most important components of GAs. We used conventional genetic operators of selection, crossover, and mutation. More precisely, we used fitness proportional selection, one-point crossover, with probability 0.75, and mutation operator with probability 0.15. 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[6]. An elitist reproduction strategy is used, where the best individual of each generation was passed unaltered to the next generation. The crossover (or recombination) operator merges the genetic information of two existing individuals (parents), picked up by selection operator, and creates two new individuals (children) called as offspring. A crossover point is randomly chosen, represented

in the Figure 1 by the dotted line and the genes to the right of the crossover point are swapped between two individuals, yielding the new offspring individuals. Note that crossover points can fall only between genes and not inside a gene.

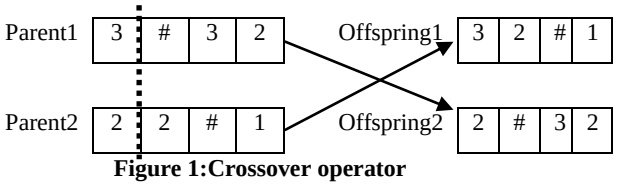

Figure 1:Crossover operator

Mutation is an operator that acts on a single individual at a time. It can introduce into an individual a gene value that is not even present in the current population. Hence, this operator maintains the diversity of gene in the population and guarantees that the search is done in the whole solution space. Mutation operator applied in this work may specialize or generalize a candidate rule by inserting or removing conditional clauses in the antecedent part of the rule. It can also mutate value of an attribute. As an example for attribute mutation as shown in Figure 2. replaces the allele '3' by the allele '2' for first attribute, removes the allele '2' for second attribute whereas it inserts the allele '2' for third attribute in the antecedent part of the rule.

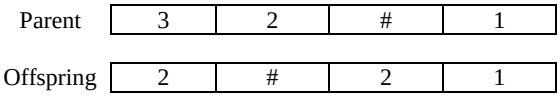

Figure 2: Mutation operator

3.3 Fitness function

Fitness function determines which chromosome should survive in the next generation and it defines the quality of chromosome as a solution to the problem. As already mentioned, the discovered small-disjunct rules are of the form, **If** P **Then** D, where

N_P Number of examples matching P.

N_D Number of examples matching D.

N_{BOTH} Number of examples matching both P and D.

N_{TOTAL} Total number of examples.

For the proposed algorithm, the Fitness function is computed as per the following formula:

$$\text{Fitness} = \frac{1}{(N_{BOTH} - (N_P \times N_D \div N_{TOTAL}))} \quad (1)$$

As an illustration of the Fitness function consider a small-disjunct rule: **If** color=blue $\wedge$ shape=brick **Then** decision=yes and the training dataset for object classification given in Table 1. For this small-disjunct rule N_P =1, N_D=4, N_{BOTH}=1 and N_{TOTAL}=7. So the Fitness is computed using the formula(1), as under:

$$\text{Fitness} = \frac{1}{(1 - (1 \times 4 \div 7))} = 2.33$$

The Fitness value favors rules which cover small number of examples in the data set being mined.

In order to evaluate the predictive accuracy of a classification model, a concept of confusion matrix is used[27]. A confusion matrix is an *d* x *d* matrix, where *d* is the number of classes in the problem at hand, that holds information about the correct and incorrect classifications made by the classification model. Figure 3 shows a confusion matrix for a two-class problem. As observed, the cells in the matrix show the number of examples correctly or incorrectly classified per class. The true positives (TP) and true negatives (TN) represent the number of examples correctly classified in the positive and negative classes, respectively. The false positives (FP) and false negatives (FN), in turn, represent the number of examples incorrectly classified as positive and negative, respectively. A simple measure of predictive accuracy (sometimes called the standard classification accuracy rate) is computed as per the following formula:

$$\text{Predictive accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (2)$$

	Predicted class	
Actual class	Yes	No
Yes	TP	FN
No	FP	TN

Figure 3: Confusion matrix

4. COMPUTATIONAL RESULTS

The performance of the suggested approach is validated on three sets of data obtained from UCI Machine Learning Repository, which is a collection of widely used benchmark and real-world data sets for data mining and KDD community[28]. The examples that had some missing values were removed from the datasets. Each GA run consisted of a population of 100 individuals evolving during 350 generations. The proposed algorithm was terminated when the maximum number of generations have reached. Each data set was randomly partitioned into two parts with 2/3 of the examples used for training and 1/3 of the examples used for testing the quality of the discovered rules. An additional restriction is often imposed in this random partitioning process. This restrict is called stratification, and it consists of imposing the restriction that in each data partition-the training and test sets of hold-out- the proportion or(relative frequency) of data instances for each class is approximately the same as in the entire original data set. In this case the procedure would be called stratified hold-out[4]. This approach is used in Section 4.1. In Section 4.2 different partitions of the test set are used to show how predictive accuracy can be affected.

4.1 Experiments

The performance of the proposed system on different datasets is demonstrated below:

4.1.1 Experiment 1

Zoo data set was used for this experiment. This dataset has 101 examples, 17 predicting attributes and a goal attribute, which can take 7 classes. The predicting attributes were nominal. Table 2 shows the discovered small-disjunct rules from the Zoo dataset.

Table 2. Result for the Zoo dataset

No.	Discovered Rules	Fitness
1	**If** demostic =0 $\wedge$ aquatic=1 **Then** Class=3	73.00
2	**If** breathes= 1 $\wedge$ demostic=1 **Then** Class =6	5.62
3	**If** demostic=0 **Then** Class=4	2.81
4	**If** fins=1 $\wedge$ cat-size=1 **Then** Class=1	2.81
5	**If** aquatic=1 $\wedge$ fins=0 **Then** Class =2	1.88
6	**If** feathers=0 **Then** Class = 5	1.63
7	**If** aquatic=1 $\wedge$ venoms=1 **Then** Class = 7	1.24

4.1.2 Experiment 2

Breast-cancer data set was used for this experiment. This dataset has 286 examples, 9 predicting attributes and a goal attribute, which can tack a two classes(201-no-recurrence-events and 85-recurrence-events). The proposed scheme would discover the following two small-disjunct rules one for each class as shown in Table 3.

Table 3. Result for the Breast-cancer dataset

No.	Discovered Rules	Fitness
1	**If** menopause= premeno ∧ node-caps= no ∧ breast=right **Then** Class = no-recurrence-events	23
2	**If** breast = left ∧ breast-quad=left-below **Then** Class = recurrence-events	17.25

4.1.3 Experiment 3

This experiment was carried out on the 1984 United States Congressional Voting Records data set. This data set has 453 examples, 16 predicting attributes and a goal attribute (Democrat and Republican). The proposed scheme would discover the following small disjunct rules (Table 4).

Table 4. Result for the Voting dataset

No.	Discovered Rules	Fitness
1	**If** el-salvador-aid=y ∧ mx-missile= y ∧ eduction-spending=n **Then** Class= Republican	19.00
2	**If** handicapped-infants=y ∧ mx-missile =n ∧ immegration=y **Then** Class= Democrat	2.38

4.2 Predictive Accuracy

In many learning tasks, the prevalence of the different classes varies significantly. For example in one of the learning tasks 90% of the training examples belong to one class and 10% to the other. If now a disjunct of only one example is used to classify unseen test examples, what accuracy we can achieved to it? It is clear that if the disjunct is associated with the majority class then the accuracy is much higher than if it is associated to the minority class. For smaller sizes, disjuncts of the minority class are notably less accurate than majority class disjuncts of the same size. The accuracy with small disjuncts predict the class of unseen examples is much lower than that of their larger brethren[29]. While disjunct size is clearly important for predicting its accuracy, as found by Holte et al.[21]. In this section we want to study how the predictive accuracy is affected by the size of test set. To study the effect of the test set size on predictive accuracy , we considered test sets of 25%, 50%, 75% and 100% of the available dataset. We run the proposed algorithm against the four partitions of each dataset to compute the overall predictive accuracy of the models. It has been shown that the predictive accuracy is increased as the number of test examples increases. So, we have noticed that the size of the test set is crucial to achieve higher predictive accuracy in the process of small disjunct rule mining. Figure 4 shows the change in predictive accuracy of the generated rules using Zoo, Breast-cancer, and Voting datasets with different test sets size.

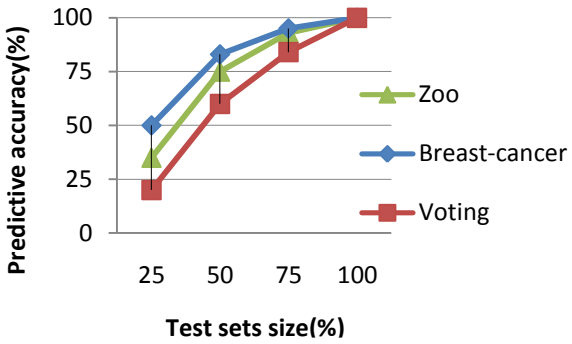

Figure 4: Change in predictive accuracy with different test sets size

5. CONCLUSION AND FUTURE WORK

The problem of how to discover good small-disjunct rules is very difficult, since these rules are error-prone due to the very nature of small disjuncts. Consequently, this paper provides insight into the role of small disjuncts in learning. The work presented in this paper has demonstrated successful application of GA for automated discovery of small-disjunct rules from large datasets. The proposed scheme has flexible chromosome encoding , appropriate crossover and mutation operators and suitable fitness function are suggested. The proposed algorithm is tested on three real world datasets and the results are quite encouraging and have established the effectiveness of the proposed algorithm. An important direction for future research is developing a method of the parallel GA to automated discovery of small-disjunct rules from large datasets.

6. REFERENCES

[1] Fayyad, U. M., Piatetsky-Sharpio, G., and Smyth, P. 1996. From mining to knowledge discovery : An overview. In: U.M. Fayyad G. Piatetsky-Sharpio, P. Smyth and R. Uthurusany (Eds.)Advances in knowledge discovery and data mining ,AAAI/MIT Press, pp. 1-34.

[2] Bramer, M. 2007. Principles of Data Mining. Springer-Verlag London Limited.

[3] Goldberg, D. E. 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesely.

[4] Frietas, A. A. 2002. Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer-Verlag Berlin Heidelberg.

[5] Quinlan, J. R. 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann.

[6] Dehuri, S., Patnaik, S., Ghosh, A., and Mall, R. 2008. Application of elitist multi-objective genetic algorithm for classification rule generation. Applied Soft Computing, vol. 8, pp. 477–487.

[7] Carvalho, D. R. and Frietas, A. A. 2002. A genetic algorithm for discovering small-disjunct rules in data mining. Applied Soft Computing, vol.2, no. 1, pp.75-88.

[8] Danyluk, A. P. and Provost, F. J. 1993. Small disjuncts in action: Learning to diagnose errors in the local loop of the telephone network. In Proceedings of 10th International Conference Machine Learning, pp.81-88,1993.

[9] Carvalho, D. R. and Frietas, A.A. 2002. A genetic algorithm with sequential niching for discovering

small-disjunct rules. In the Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002), New York, pp. 1035-1042.

[10] Yogita, Saroj, and Kumar, D. 2009. Rule +Exceptions: Automated discovery of comprehensible decision rules. IEEE International Advance Computing Conference(IACC2009),Patiala, India, pp. 1479-1483.

[11] Carvalho, D. R. and Freitas, A. A. 2000. A genetic algorithm-based solution for the problem of small disjuncts. Principles of data mining and knowledge discovery. In Proceedings of the 4th European Conference, PKDD-2000, Lyon, France. Lecture Notes in Artificial Intelligence 1910, Springer, Berlin, pp. 345-352.

[12] Carvalho, D. R. and Freitas, A. A. 2000. A hybrid decision-tree/genetic algorithm for coping with the problem of small disjuncts in data mining. In Proceedings of the 2000 Genetic and Evolutionary Computation Conference (Gecco-2000), Las Vegas, NV, USA, pp. 1061–1068.

[13] Sarkar, B. K., Sana, S. S., and Chaudhuri, K. 2012. A genetic algorithm-based rule extraction system", Applied Soft Computing, vol.12, pp. 238-254.

[14] Bharadwaj, K. K. and Al-Maqaleh, B. M. 2006. Evolutionary approach for automated discovery of augmented production rules. International Journal of Computational Intelligence, vol. 3, Issue 4, pp. 267-275.

[15] Al-Maqaleh, B. M. and Bharadwaj, K. K. 2007. Evolutionary approach to automated discovery of censored production rules with fuzzy hierarchy. In Proceedings of the International Conference on Data Mining and Applications (ICDMA'2007), Hong Kong, China, vol. 1, pp. 716-721.

[16] Dehuri, S. and Mall, R. 2006. Predictive and comprehensible rule discovery using a multi objective genetic algorithms. Knowledge Based Systems, vol. 19, pp. 413-421.

[17] Liu, B., Hu, M. and Hsu, W. 2000. Multi-level organization and summarization of the discovered rules. In Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovered and Data Mining (KDD-2000), ACM Press, pp. 208–220.

[18] Weiss, G. M. and Hirsh, H. 2000. A quantitative study of small disjuncts. In Proceedings of the of 17th National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, pp. 665– 670.

[19] Gomes, A. K. 2007. Small disjuncts grouping by rule coverage and accuracy measures. In 7th IEEE International Conference on Intelligent Systems Design and Applications, pp. 412-415.

[20] Weiss, G. M. 1998. The problem with noise and small disjuncts. In Proceedings of the International Conference on Machine Learning (ICML-98), Morgan Kaufmann, Los Altos, CA, pp. 574–578.

[21] Holte, R. C., Acker, L. E., and Porter, B. W. 1989. Concept learning and the problem of small disjuncts. In Proceedings of IJCAI – 89, pp. 813-818.

[22] Weiss, G. W. 1995. Learning with rare cases and small disjuncts. In Proceedings of the 12th International Conference on Machine Learning (ICML-95), Morgan Kaufmann, Los Altos, CA, pp. 558–565.

[23] Ting, K. M. 1994. The problem of small disjuncts: its remedy in decision trees. In Proceedings of the 10th Canadian Conference on AI, pp. 91-97.

[24] Saroj, R. and Bharadwaj, K. K. 2009. Discovery of exceptions: A step towards perfection. Third IEEE International Conference on Network and System Security, Banaras Hindu University, India, pp.540-545.

[25] Al-Maqaleh, B. M. 2012. Genetic algorithm approach to automated discovery of comprehensible production rules. In the proceedings of the 2nd International Conference on Advanced Computing & Communication Technologies(ACCT2012), Rohtak, India, pp. 69-71.

[26] Thornton, C.J. 1992. Techniques in Computational Learning-An Introduction. London: Chapman & Hall.

[27] Witten, I.H., Frank, E. 2005 Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, 2nd edition. Morgan Kaufmann.

[28] UCI Repository of Machine Learning Databases, Department of Information and Computer Science University of California, 1994. [http://www.ics.uci.edu/ ~mlearn/MLRepositry.html].

[29] Quinlan, J. R. 1991. Improved estimates for the accuracy of small disjuncts. Journal of Machine Learning, Kluwer Academic Publishers Hingham, MA, USA, vol. 6(1), pp. 93-98.