Volume 4, Issue 7, July 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Method Based on Genetic Algorithm for Discrimination Discovery

Saheeda A* Computer Science, MG University India
Shanavas K A Information Technology, MG University India
Sandra G Computer Science, MG University India

Abstract- Discrimination is an important issue in social and economical science. It involves the group's initial reaction or interaction, influencing the individual's actual behaviour towards the group or the group leader, restricting members of one group from opportunities or privileges that are available to another group, leading to the exclusion of the individual or entities based on logical or irrational decision making (for e.g. race, religion, sex etc).For discovering the discrimination of a particular group on a particular context based on a decision can be best expressed by classification rules. There are several methods in data mining for discovering and preventing both direct and indirect discriminations. In the existing works, these classification rules must be manually specified for each type of discrimination. In this paper, a method based on genetic algorithm is proposed for automatically generating classification rules from the history of the dataset. In the proposed system the fitness function is calculated in a different way. As the name itself, new rules can be generated from history and create better matches for classification rules through several steps. Experimental results show that proposed method gives better results. It avoids the difficulty of manually creating the rules.

Keywords— Data Mining, Discrimination Discovery, Genetic Algorithm, Discrimination Prevention, Fitness Function

I. INTRODUCTION

In a social sense, discrimination refers to an action based on prejudice resulting in unfair treatment of people, where the distinction between people is operated on the basis of their membership to a category or minority, without regard to individual merit or circumstances. Examples of social discrimination include racial/ethnic, religious, sexual orientation, disability, and age-related discrimination; People do not like to be discriminated based on their race, ethnicity, religion, nationality, gender, sexuality, disability, marital status, genetic features, language and age. These types of discrimination occur in various situations like employment and training, public services, education and health care; credit and insurance etc. Many people may not mind other people knowing about their ethnic origins, but they would strongly be denied a credit or a grant if their ethnicities were part of that decision. With the advent of automatic decision support systems, such as credit scoring systems, socially sensitive decisions may be taken by automatic systems, e.g., for screening or ranking applicants for a job position, to a loan and so on. These types of decision can be taken by generating classification rules from trained dataset in data mining..Discrimination discovery in databases consists in the actual discovery of discriminatory situations and practices hidden in a large amount of historical decision records. So we need technologies for discovering and preventing discriminations.

Discriminations are of two types. In direct discrimination, one person is treated less favourably on the grounds of sex than another is, has been or would be treated in a comparable situation. In indirect discrimination, an apparently neutral provision, criterion or practice would put persons of one sex at a particular disadvantage compared with persons of the other sex, unless that provision, criterion or practice is objectively justified by a legitimate aim, and the means of achieving that aim are appropriate and necessary.Anti discrimination laws [4] are developed in some of the countries. So there is a need for preventing both direct and indirect discrimination. There are several research papers are available in which some of them deals with the discovery and measurement and some of them deals with prevention of discrimination. In the existing work for prevention of discrimination, all the algorithms create classification rules for discrimination discovery manually. This paper automates the work for generating rules for discrimination discovery.

II. RELATED WORKS

Work of Pedreschi et al. [1], [4] provide the discovery of discrimination based on mining, classification rules that give a legal definition of discrimination which is based on some qualitative measure. Then check whether evidence of discrimination can be found in a given set of decisions [2], by measuring the degree of discrimination of a rule got from an expert's theory– e.g., a doubtful pattern to verify by an anti-discrimination body is interested in. The next natural step is to repeat such a procedure for all the classification rules that appear from the historical data, thus publishing all the discriminatory patterns hidden in the data.

In Dino Pedreschi ‘s work [3] the discrimination analysis relies on the definition of the groups of interest, on a measure of discrimination for a classification rule, computable starting from frequencies extracted from the training set, and on a few legal principles, that can be formalized through meta-rule deductions over the set of extracted rules. Discrimination Prevention [7] of Sara Hajian consists of producing patterns that do not lead to discriminatory decisions, even if the original training data sets are biased. There are three methods for transforming data .But Sara Hajian focus on pre-processing approach. And specifies some of the disadvantages of this method. Earlier works only deals with direct discrimination; they do not consider indirect discrimination. So in [7] proposes some data transformation methods that consider both direct and indirect discrimination. In these methods we give input as classification rules. So for each time we generate the rules and give this as input. To overcome this limitation, this paper proposes a new method based on genetic algorithm [8] to generate these rules directly from the history of training data set. So it reduces the difficulty of manually creating the rules for each data item.

The rest of this paper is organized as follows. Section II introduces related works for this paper III some basic that are used throughout the paper. Section III describes a genetic algorithm for extracting classification rules. Section IV shows the tests we have performed to. Finally, Section V summarizes conclusion.

III. SOME BASIC CONCEPTS

For mining the association rules some of the definitions are used.

- A data set is a collection of data objects (records) and their attributes. Let DB be the original data set.
- An item is an attribute along with its value, e.g., Race= black.
- An item set, i.e., X, is a collection of one or more items, e.g., {Foreign worker =Yes, city=NYC}.A classification rule is an expression X → C, where C is a class item (a yes/no decision), and X is an item set containing no class item, e.g., {Foreign worker = Yes; City = NYC} Hire → no. X is called the premise of the rule.
- The support [5] of an item set, Supp (X), is the fraction of records that contain the item set X. We say that a rule X → C is completely supported by a record if both
  X and C appear in the record.
- The confidence [5] of a classification rule, conf (X→ C), measures how often the class item C appears in records that contain X. Hence, if supp(X) > 0 then conf(X → C) =supp(X, C)/ supp(X) : Support and confidence range over [0,1].
- A frequent classification rule is a classification rule with the support and confidence greater than respective specified lower bounds.
- The negated item sets, i.e., ¬X is an item set with the same attributes as X, but the attributes in X take any value except those taken by attributes in X

3.1 Potentially Discriminatory and Non-discriminatory Classification Rules

Let DIs be the set of predetermined discriminatory items in DB (e.g., DIs = {Foreign worker = Yes, Race = Black, Gender =Female}). Frequent classification rules in the FR fall into one of the following two classes:

- A classification rule X →C is potentially discriminatory (PD) when X = A, B to A is a subset of DIs, a nonempty discriminatory item set and B a non-discriminatory item set.
- A classification rule X → C is potentially non-discriminatory (PND) when X = D, B is a non-discriminatory item set.

A PD rule could probably lead to discriminatory decisions. A PND rule could lead to discriminatory decisions in combination with some background knowledge. Hence, measures are needed to quantify both direct and indirect discrimination potentials.

Discrimination discovery: - Direct and indirect discrimination discovery [5] includes identifying α-discriminatory rules and redlining rules. First, based on predetermined discriminatory items in DB, frequent classification rules are divided in two groups: PD and PND rules. Second, direct discrimination is measured by identifying α-discriminatory rules among the PD rules using a direct discrimination measure and a discriminatory threshold. Third, indirect discrimination is measured by identifying redlining rules among the PND [6] rules combined with background knowledge, using an indirect discriminatory measure, and a discriminatory threshold. Direct discrimination is modelled through potentially discriminatory rules, which are classification rules A, B → C that contain potentially discriminatory item sets A in their premises. We show in 3.1 and 3.2 that there is always a unique split of the premise into a PD part and a non PD part. A PD rule does not necessarily provide evidence of discriminatory actions. In order to measure the “disproportionate burdens” that a rule imposes, the notion of α –protection [7] is introduced as a measure of the discriminatory power of a PD classification rule. The idea is to define such a measure as the relative gain in confidence of the rule due to the presence of the discriminatory item sets. The parameter is the key for tuning the desired level of protection against discrimination. PD classification rules are extracted from a dataset containing discriminatory item sets.

The algorithms for data transformation for preventing both direct and indirect discrimination specified in the [8] .In this paper, we specify a method based on genetic algorithm as input to the prevention algorithms to extract the rules automatically from history of the dataset.

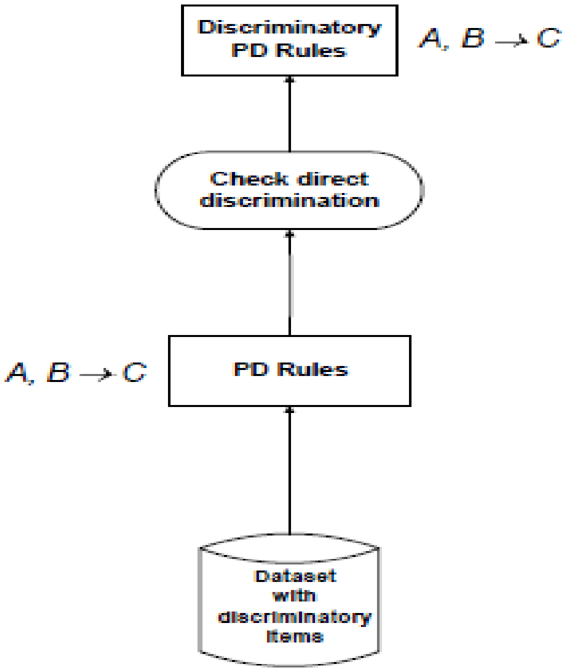

Fig. 3.1 Modelling the process of direct discrimination control

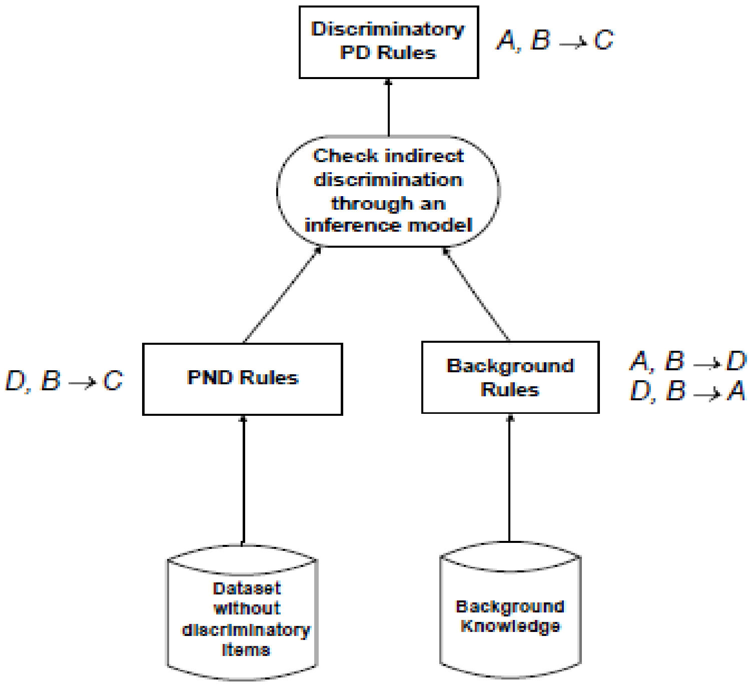

Fig. 3.2 Modelling the process of indirect discrimination control

V. A METHOD BASED ON GENETIC ALGORITHM

In a genetic algorithm [8], a population of candidate solutions (called individuals) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its attributes) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible.

The evolution usually starts from a population of randomly generated individuals (candidate rule) and is an iterative process, with the population in each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's data item is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

A typical genetic algorithm requires:

1. a genetic representation of the solution domain,
2. a fitness function to evaluate the solution domain.

A standard representation of each candidate solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. Once the genetic representation and the fitness function are defined, a GA proceeds to initialize a population of solutions and then to improve it through repetitive application of the mutation, crossover, inversion and selection operators. Steps in GA are shown in Fig.4.1

Initialization

Initially many individual solutions are (usually) randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, allowing the entire range of possible solutions (the search space). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

Selection

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as the former process may be very time-consuming.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. In some problems, it is hard or even impossible to define the fitness expression; in these cases, a simulation may be used to determine the fitness function value of a phenotype, or even interactive genetic algorithms are used. When using this kind of measure, discovered rule is often considered interesting in the sense of being novel and/or surprising for the user—when it contradicts the previous knowledge or expectation of the user. Here the fitness function is based on a rank. It can be calculated by taking the ratio of number of matched records from history and rule size.

$$\text{Rank} = \frac{\text{Number of Matched records from history}}{\text{rule size}}$$

Genetic operators

For each new solution, to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated. Although reproduction methods that are based on the use of two parents are more "biology inspired", some research suggests that more than two "parents" generate higher quality rules.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure in the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

It is worth tuning parameters such as the mutation probability, crossover probability and population size to find reasonable settings for the problem class being worked on. A very small mutation rate may lead to genetic drift. A recombination rate that is too high may lead to premature convergence of the genetic algorithm. A mutation rate that is too high may lead to loss of good solutions unless there is an elitist selection. There are theoretical, but not yet practical upper and lower bounds of these parameters that can help guide selection.

Termination

This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

- A solution is found that satisfies minimum criteria
- Fixed number of generations reached
- Allocated budget (computation time/money) reached
- The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results.
- Manual inspection
- Combinations of the above

Finally the output of this genetic algorithm is given as input to the discrimination prevention algorithms in [7] and prevents both direct and indirect discrimination. The experimental results show that the proposed algorithm is quite successful in both goals of discovering discrimination and removing discrimination.

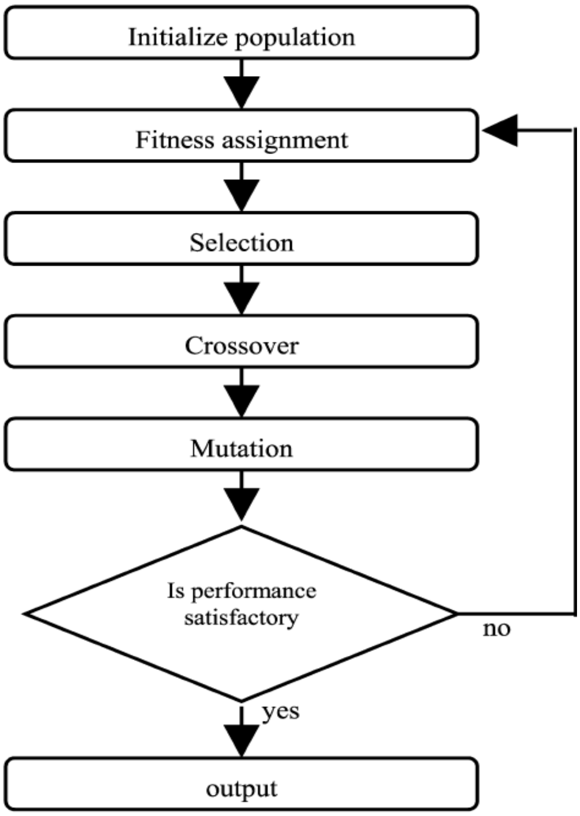

Fig.4.1 Steps in genetic algorithms

VII. EXPERIMENTAL RESULTS

The algorithm implemented in J2SDK1.5 The tests were performed on an 1.1GHz Pentium machine, equipped with 1GB of RAM, and running under Windows 7 Professional. Genetic algorithm gives the best matched rules from the history through the above steps. Cross over and mutation steps produce higher quality rules. When the number of iteration increases, then match value also increases, that means, we get high match value as shown in Fig.7.1. These rules are given as input to discrimination prevention algorithms and can prevent both direct and indirect discrimination.

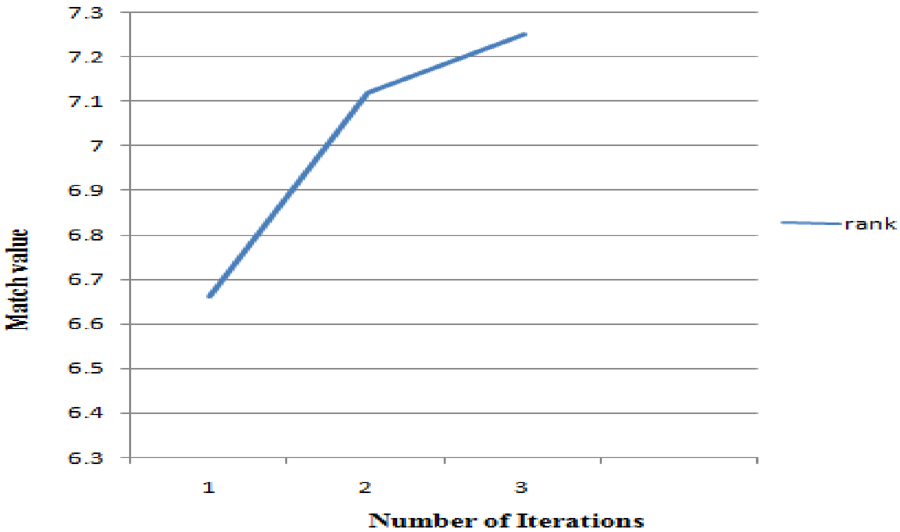

Fig.7.1 Performance graph

VIII. CONCLUSION

The purpose of this paper was to develop a new discrimination prevention methodology that can prevent direct discrimination, indirect discrimination or both at the same time using genetic algorithm. To attain this objective, the first step is to identify the groups in which people directly or indirectly discriminated in the decision-making processes; then measure those discrimination the second step is to remove all those discriminatory biases.

REFERENCES

[1] D. Pedreschi, S. Ruggieri, and F. Turini,“Discrimination-Aware Data Mining,” Proc. 14th ACM Int’l Conf. Knowledge Discovery and Data Mining (KDD ’08), pp. 560- 568, 2008.

[2] D. Pedreschi, S. Ruggieri, and F. Turini,“Measuring Discrimination in Socially-Sensitive Decision Records,” Proc. Ninth SIAM Data Mining Conf. (SDM ’09), pp. 581-592, 2009.

[3] D. Pedreschi, S. Ruggieri, and F. Turini, “Integrating Induction and Deduction for Finding Evidence of Discrimination,” Proc. 12th ACM Int’l Conf. Artificial Intelligence and Law (ICAIL ’09), pp. 157- 166, 2009.

[4] S. Ruggieri, D. Pedreschi, and F. Turini, “Data Mining for Discrimination Discovery,” ACM Trans. Knowledge Discovery from Data, vol. 4, no. 2, article 9, 2010.

[5] S. Hajian, J. Domingo - Ferrer, and A. Martı nez - Balleste, “Discrimination Prevention in Data Mining for Intrusion and Crime Detection,” Proc. IEEE Symp. Computational Intelligence in Cyber Security (CICS ’11), pp. 47-54, 2011.

[6] S. Hajian, J. Domingo-Ferrer, and A. Martınez-Balleste, “Rule Protection for Indirect Discrimination Prevention in Data Mining,” Proc. Eighth Int’l Conf. Modeling Decisions for Artificial Intelligence (MDAI ’11), pp. 211-222, 2011.

[7] Sara Hajian and Josep Domingo-Ferrer, Fellow, IEEE “ A Methodology for Direct and Indirect Discrimination Prevention in Data Mining”: IEEE transactions on knowledge and data engineering, vol. 25, no. 7, July 2013

[8] S. Dehuri , S. Patnaik a, A. Ghosh b, R. Mall “Application of elitist multi-objective genetic algorithm for classification rule generation” Science Direct,Applied Soft Computing 8 (2008) 477–487

