DCUBE: Discrimination Discovery in Databases

Salvatore Ruggieri Dino Pedreschi Franco Turini

Dipartimento di Informatica, Università di Pisa
L.go B. Pontecorvo 3, 56127 Pisa, Italy
{ruggieri,pedre,turini}@di.unipi.it

ABSTRACT

Discrimination discovery in databases consists in finding unfair practices against minorities which are hidden in a dataset of historical decisions. The DCUBE system implements the approach of [5], which is based on classification rule extraction and analysis, by centering the analysis phase around an Oracle database. The proposed demonstration guides the audience through the legal issues about discrimination hidden in data, and through several legally-grounded analyses to unveil discriminatory situations. The SIGMOD attendees will freely pose complex discrimination analysis queries over the database of extracted classification rules, once they are presented with the database relational schema, a few adhoc functions and procedures, and several snippets of SQL queries for discrimination discovery.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

General Terms

Algorithms, Legal Aspects

Keywords

Discrimination, Classification Rules

1. INTRODUCTION

Civil right laws worldwide prohibit discrimination on the basis of race, color, religion, nationality, sex, marital status, age and pregnancy in a number of settings, including: credit and insurance; sale, rental, and financing of housing; personnel selection and wages; access to public accommodations, education, nursing homes, adoptions, and health care [2, 6]. A general principle is to consider group underrepresentation as a quantitative measure of the qualitative requirement that people in a group are treated "less favorably" than others, or such that "a higher proportion of people without the attribute comply or are able to comply" to a qualifying criterium. With the advent of automatic decision support systems, such as credit scoring systems, the ease of data collection opens several challenges to data analysts for the fight against discrimination. Discrimination discovery in databases consists in the actual discovery of discriminatory situations and practices hidden in a large amount of historical decision records. The process of data analysis must then be supported by tools that implement legally-grounded measures and reasonings.

The first approach to discrimination discovery from a computer science perspective is based on extracting and reasoning about classification rules [4, 5]. The various concepts and analyses, originally implemented as a stand-alone program for achieving the best performances, have been re-designed around an Oracle database, storing extracted itemsets and rules, and a collection of functions, procedures and snippets of SQL queries that implement the various legal reasonings for discrimination analysis. The resulting implementation, called DCUBE, can be accessed and exploited by a wider audience if compared to a stand-alone monolithic application. Discrimination discovery is an interactive and iterative process, where analyses assume the form of deductive reasoning over extracted rules. An appropriately designed database, with optimized indexes, functions and query snippets, can be welcome by a large audience of users, including owners of socially-sensitive decision data, government anti-discrimination analysts, technical consultants in legal cases, researchers in social sciences, economics and law.

We describe the architecture of DCUBE, and a demonstration which: (i) introduces the audience to the issue of discrimination discovery, by making them aware of the legal issues (their own!) data can hide, and to an approach for discrimination analysis; (2) guides the audience through the processes for discovering direct discrimination, affirmative actions, indirect discrimination, favoritism and respondent argumentation; (3) allows the participants to directly interact by posing specific queries, through standard SQL, over the DCUBE database. Case studies for the demonstration include a few publicly available datasets on loans, such as the German credit dataset[1] and the PKDD 1999 financial dataset[2].

2. A 5 MINUTES TUTORIAL

The demonstration begins with a brief introduction to the concepts and methods for discrimination discovery following the approach of [4, 5].

[1] `http://archive.ics.uci.edu/ml`
[2] `http://lisp.vse.cz/challenge`

SIGMOD'10, June 6–11, 2010, Indianapolis, Indiana, USA.

Attributes
on personal properties: checking account status, duration, savings status, property magnitude, type of housing *on credits:* credit history, credit request purpose, credit request amount, installment commitment, existing credits, other parties, other payment *on employment:* job type, employment since, number of dependents, own telephone *on personal status:* personal status and gender, age, resident since, foreign worker
Decision
The decision attribute is "class", with values good (grant credit) and bad (deny credit)
Potentially Discriminatory Items
`personal_status=female div/sep/mar` (female non-single) `age=(52.6-inf)` (senior people) `job=unemp/unskilled non res` (unskilled or unemployed) `foreign_worker=yes` (foreign workers)
Favored Groups Items
`personal_status=male single` (male and single) `age=(41.4-52.6]` (age around '40s)

Table 1: The German credit case study

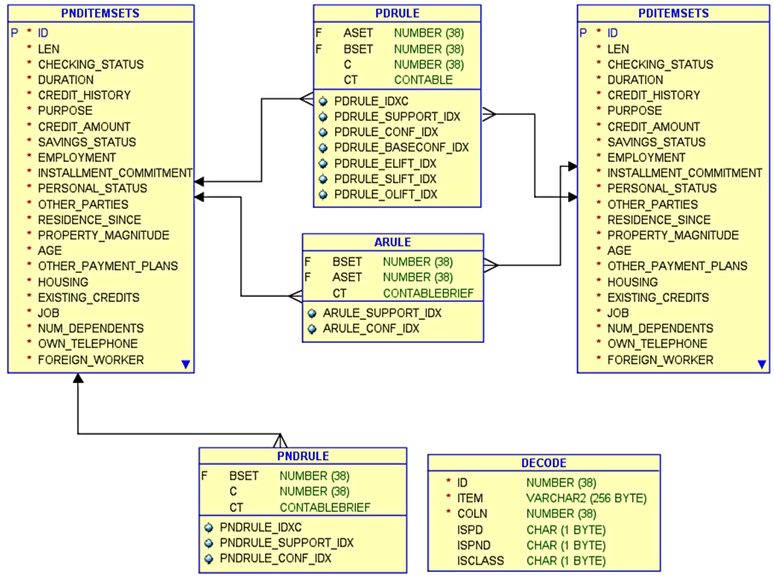

Figure 1: Database schema generated by DCUBE for the German credit case study

Classification and association rules for discrimination discovery are extracted from a dataset of historical decision records, namely a database table with attributes used for a decision and the decision outcome itself. Protected by-law groups are denoted by a collection of itemsets called potentially discriminatory (PD), while contexts where discrimination may occur are described by potentially non-discriminatory (PND) itemsets. Consequently, the extracted classification rules are partitioned into PD rules, containing a PD itemset, and PND rules, with no PD itemset.

PD rules of the form $\mathbf{A}, \mathbf{B} \rightarrow \mathbf{C}$, where $\mathbf{A}$ is a protected by law group and $\mathbf{B}$ is a context, can be searched for cases of possible discrimination where some quantitative measure of discrimination exceeds a legally-grounded threshold. A few measures of discrimination based on existing laws, jurisprudence and social studies are introduced in [4, 5] and defined over the 4-fold contingency table of a PD rule, including: extended lift, selection lift, odds lift. Tests of statistical significance of the various measures are introduced in [4]. When $\mathbf{C}$ is the negative class, e.g., deny credit, this search unveils discrimination against protected-by-law groups.
When $\mathbf{C}$ is the positive class, e.g., grant credit, this search unveils affirmative actions, namely policies favoring minorities, sometimes encouraged or enforced by laws. Finally, by defining PD itemsets to denote favored groups, the same process of analysis can support the discovery of favoritism.

PND rules $\mathbf{D}, \mathbf{B} \rightarrow \mathbf{C}$, where both $\mathbf{D}$ and $\mathbf{B}$ are PND itemsets, are subject to analysis as well, since they may reveal apparently neutral practices, known as indirect discrimination, whose effects on protected-by-law groups are the same of some PD rule $\mathbf{A}, \mathbf{B} \rightarrow \mathbf{C}$ – which may not be extracted because data does not contain the attributes in $\mathbf{A}$, e.g., the race of an applicant to a loan. To discover indirect discrimination, some additional background knowledge must be exploited, such as census data on the distribution of population over the territory. [5] assumes that background knowledge is provided as a set of association rules of the form $\mathbf{B} \rightarrow \mathbf{A}$, where $\mathbf{B}$ is a PND and $\mathbf{A}$ is a PD itemset.

3. DEMONSTRATION SCENARIO

The "5 minutes tutorial" on discrimination discovery provides the audience with the basic definitions for understanding the DCUBE functionalities. The system architecture of DCUBE is presented next, as discussed in the Section 4.

The demonstration then proceeds by running the rule extraction phase of DCUBE on the case study datasets (see Table 1). The populated database is navigated to make the participants acquainted with its schema (see Fig. 1) The user interface of DCUBE is integrated within Oracle SQL Developer[3], hence the demonstration will run within a single GUI. The coding of itemsets and rules in the database is presented together with Oracle user defined functions for coding/decoding and for splitting itemsets into their PD and PND parts. Also, the Oracle user defined type modelling 4-fold contingency tables is presented together with sample usages of its methods for computing a few legally-grounded measures of discrimination. Finally, utility views, function indexes, and bitmap join indexes defined on the database to optimize query performances are briefly surveyed.

The main part of the demonstration consists of presenting snippets of SQL queries, produced by DCUBE (see Fig. 2), that provide an answer to typical discrimination discovery issues. In increasing level of complexity, we will deal with:

1. **Direct discrimination discovery**. It consists of studying the distribution of a discrimination measure over the set of PD classification rules, with the intent to select and interpret the rules with the highest measure values. The users are allowed to ask queries such as *"How much have women been under-represented in obtaining the loan?"* or *"List under which conditions blacks were suffering a selection lift higher than 1.25[4] in our recruitment data"*. While DCUBE comes with

[3] `http://www.oracle.com/technology/products/database/sql_developer`

[4] This threshold is known as the "four-fifths" rule [6](d).

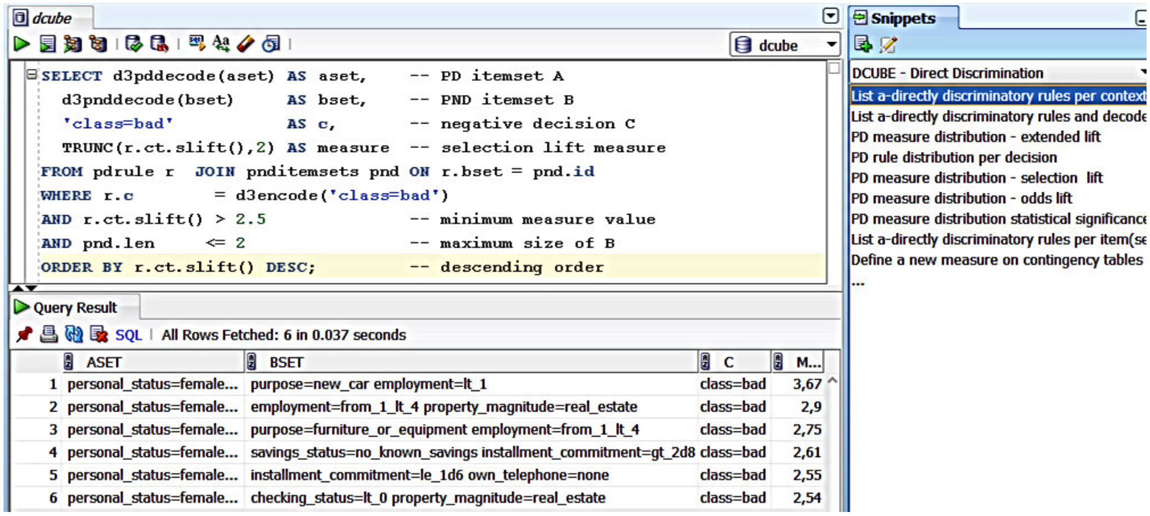

Figure 2: SQL query snippet screenshot

a few legally-grounded measures predefined, we show how the user can easily define new measures by adding methods to an Oracle user defined data type.

2. **Affirmative actions and favoritism discovery**. Affirmative action discovery is shown to be a variant of direct discrimination discovery. It can answer questions such as *"List cases where our university admission policies actually favored blacks"*. On the contrary, favoritism discovery requires to re-process the rule extraction phase, since the PD itemsets now include suspiciously favored groups. It can answer questions such as *"Under which conditions white males are given the best mortgage rate in comparison to the average?"*

3. **Indirect discrimination discovery**. This is an inferential problem, where an explicit PD rule has to be derived starting from PND rules and background knowledge. We show how to add background knowledge into the DCUBE database in the form of association rules of the form $\mathbf{B} \rightarrow \mathbf{A}$, such as *"If resident in Indianapolis then black (25.4%)*[5]*"*. Moreover, for testing and validation purposes, DCUBE allows the user to simulate the availability of a large set of background rules under the assumption that the dataset contains the PD items. In such a cases, all association rules of the form $\mathbf{B} \rightarrow \mathbf{A}$ with a specified minimum support are extracted from the dataset under analysis. The participants will be presented with two inference strategies introduced in [4, 5] as a means for posing indirect discrimination discovery questions in:

   - redlining analysis, such as *"I don't have the race attribute in my data, but have the ZIP of residence. By adding background knowledge on the distribution of race over ZIP codes, infer cases where ZIP actually disguises race discrimination."*
   - discrimination through favoritism analysis, such as *"My decision support system does not take into account the status of being a foreign worker when denying credit to an applicant. Are there any cases where actually such a statement has been deceived by disproportionately granting credit to local workers?"*

For the technically interested audience, the execution plans of the SQL query snippets can be presented and discussed.

4. DCUBE ARCHITECTURE

DCUBE supports the discrimination discovery process of Fig. 3. The user starts the DCUBE wizard through the Oracle SQL Developer GUI. The wizard allows for selecting the following inputs: (1) a relational table, view or SQL query from a JDBC data source, or from a CSV text file; (2) a minimum support threshold; (3) a list of PD items – with all other items treated as PND; (4) a list of class items; (5) a target Oracle schema. Additional inputs constraint the set of classification rules to be extracted by setting: the maximal size of a frequent itemset; the maximum support threshold; the maximal similarity threshold between items, after which two or more similar items are merged. Based on those inputs, DCUBE proceeds with the phases of mining, loading and querying.

Mining. Data from the input table (1) is fetched and frequent itemsets [3] are extracted for the specific minimum support (2). Any system from the Frequent Itemset Mining Implementations repository[6] can be plugged in the DCUBE system. By default, the PATTERNIST algorithm [1] is adopted, which allows for specifying several types of constraints over frequent itemsets. Starting from the extracted frequent itemsets, an implementation of the procedures designed in [4, 5] is adopted to compute classification rules whose consequent is in the list (4); to split the PD part of the antecedent of a rule accordingly to the list (3); to extract

[5]This sample rule is derived from the 2000 Census of Population and Housing (`http://factfinder.census.gov`), which provides data at the detailed level of Zip Code Tabulation Areas.

[6]`http://fimi.cs.helsinki.fi`

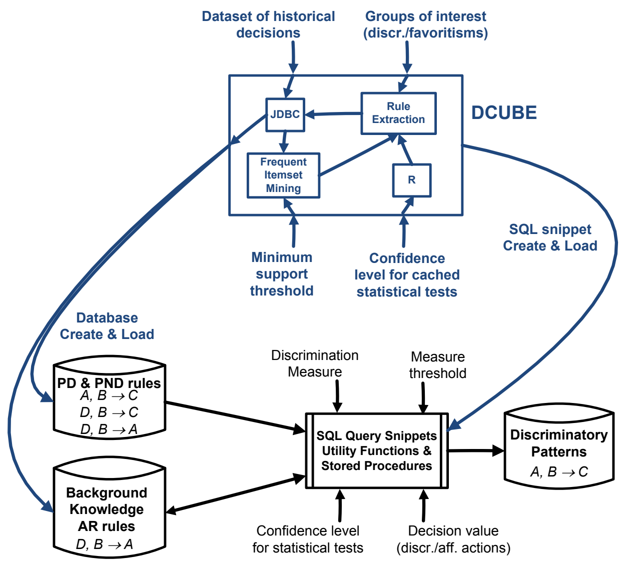

Figure 3: Analysis process supported by DCUBE

the 4-fold contingency table of the PD classification rules. Also, association rules of the form $\mathbf{B} \rightarrow \mathbf{A}$ with minimum support (2) are extracted starting from frequent itemsets as a means to simulate the availability of background knowledge. Finally, a module written in the R[7] statistical software language is part of the DCUBE system for computing confidence intervals of the discrimination measures. Computationally expensive calculations of confidence intervals (see [4] for a discussion) are cached.

Loading All extracted rules are loaded in the target schema (5), whose structure is generated by DCUBE starting from the input table columns (see Fig. 1). Basically, integer codes are assigned to items, to PD itemsets and to PND itemsets, with tables DECODE, PDITEMSETS and PNDITEMSETS storing them respectively. Utility stored procedures for coding, decoding and splitting an itemset into its PD and PND parts are created as well. PD rules are stored in the PDRULE table together with their 4-fold contingency tables, which is modelled by an Oracle user defined object data type called CONTABLE. Measures of discrimination are implemented as methods over such a data type. Similarly, PND rules are loaded in the PNDRULE table, and association rules simulating background knowledge in the ARULE table. The user can add her own background knowledge in the form of association rules by inserting them into the ARULE table. Utility stored procedures are created to help the user in this task. Also, utility views are defined by DCUBE, such as the SUBITEMSETS view that relates an itemset code to the codes of its subitemsets.

Querying The DCUBE wizard generates two dozens of optimized SQL query snippets, and store them in the SQL Developer snippets repository. Query snippets model typical queries that an anti-discrimination analyst may be interested in. Intuitively, the user drags and drops a query snippet from the list of generated ones to the SQL query editor, modifies any parameter of the query (e.g., the minimum discrimination measure value to search rules for), and then runs the query against the database (see Fig. 2). This approach is effective both for the novice user, who is guided in her first steps, and for the experienced user, who can augment the set of snippets generated by her own ones. The querying phase can be iterated by the analyst to explore the search space by varying: the contexts of possible discrimination, the reference formal measures of discrimination, the minimum measure threshold for PD rules, the minimum confidence level in statistical significance of the measure values, the inference strategies for indirect discrimination, the subgroups of minorities or favored people to look at.

The whole mining-loading-querying process is iterated as far as the analysis require a re-extraction of classification rules, as when considering a different dataset, a lower minimum support threshold, or a different group of people to reason about.

[7] `http://www.R-project.org`

5. CONCLUSIONS

DCUBE is an analytical tool supporting the interactive and iterative process of discrimination discovery. The intended users of DCUBE include: owners of socially sensitive decision databases, anti-discrimination authorities and auditors, researchers in social sciences, economics and law. DCUBE re-implements the theoretical foundations presented in [4, 5] by centering the analysis phase around an Oracle database. This approach has the following benefits: minimization of the learning gap for data analysts already acquainted with relational databases; partial automation of the process of discrimination discovery through routinely scheduled executions of rule extraction and query snippets; efficiency of the query snippet executions; extensibility of the analysis, e.g., by defining additional measures and query snippets. An on-line demo is available from the DCUBE home page:

`http://kdd.di.unipi.it/dcube`

6. REFERENCES

[1] F. Bonchi and C. Lucchese. Extending the state-of-the-art of constraint-based pattern discovery. *Data & Knowledge Engineering*, 60(2):377–399, 2007.

[2] European Union Directives. (a) Racial Equality, (b) Employment Equality. `http://ec.europa.eu/-employment_social/fundamental_rights`.

[3] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: Current status and future directions. *Data Mining and Know. Disc.*, 15(1):55–86, 2007.

[4] D. Pedreschi, S. Ruggieri, and F. Turini. Measuring discrimination in socially-sensitive decision records. In *Proc. of SDM 2009*, pages 581–592. SIAM, 2009.

[5] S. Ruggieri, D. Pedreschi, and F. Turini. Data mining for discrimination discovery. *ACM Trans. on Know. Disc. from Data*, 2010. To appear. Preliminary version appeared as Discrimination-aware data mining in *Proc. of KDD 2008*, pages 560–568.

[6] U.S. Legislation. (a) Equal Credit Opportunity Act, (b) Fair Housing Act, (c) Intentional Employment Discrimination, (d) Equal Pay Act, (e) Pregnancy Discrimination Act. `http://www.usdoj.gov`.