CS 500, Fundamentals of Databases, Winter 2018

Homework 6: Mining Fair Classification Rules

Due at 5pm on Monday, March 19, 2018

NO LATE SUBMISSIONS WILL BE ACCEPTED

Description

This assignment is about responsible data mining. You are asked to analyze a post-processed version of the ProPublica dataset that is provided for you, by writing SQL queries.

Details and Grading

This assignment is worth 5% of the over-all course grade. If this assignment is submitted late, you will receive no credit.

This assignment is to be completed individually. Please consult the course syllabus for a description of our academic honesty policy.

Your SQL queries should run in PostgreSQL. I will grade your submission by executing your queries on tux. You will not receive full credit if I cannot run your queries.

Submission instructions

Submit your assignment to gitlab on gitlab.cci.drexel.edu. I assume that you already followed the steps to create your git repository. I will refer to the root directory of your git repository as `$GIT_HOME`. Create a directory called `cs500-hw6` (case-sensitive, use exactly this name) under `$GIT_HOME`:

```
mkdir $GIT_HOME/cs500-hw6
cd $GIT_HOME/cs500-hw6
```

You should submit your work in one file, `Fairness.sql`. Do not rename the file. Place the file into your `$GIT_HOME/cs500-hw6` directory. Submit your assignment as follows:

```
git add Fairness.sql
git commit -m 'homework 6'
git push
```

You may submit multiple times before the deadline, only your last submission committed before the deadline will be graded.

Assignment specification

Read the investigation by Pro Publica on racial bias in criminal sentencing at
https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Download the post-processed criminal sentencing dataset, the database creation script, and the solution template from
https://www.cs.drexel.edu/~julia/cs500/documents/code/PP.csv
https://www.cs.drexel.edu/~julia/cs500/documents/code/Load_PP.sql
http://cs.drexel.edu/~julia/cs500/documents/code/Fairness.sql

Load the tables and the data into your PostgreSQL schema by executing:

```
julia_cs500> \i Load_PP.sql
```

The dataset consist of one table: Pro_Publica (uid, attr, val). Check that the table has 29,375 rows by executing `select count(*) from Pro_Publica.`

We will be working with classification association rules (CARs) in this assignment. We covered CARs in class; see slides for Lecture 8. We implemented part of CAR mining in Fairness.sql, which is provided to you as part of this assignment. To execute our CAR mining code, run Fairness.sql:

```
julia_cs500> \i Fairness.sql
```

Study the queries in Fairness.sql and their output carefully, to understand what is computed and how. Note that:

- CARs with one attribute-value in the body are in the table R2, while CARs with 2 attributes in the body should go into the table R3 (you'll compute those in Part 1 of the assignment).

- CARs are mined with respect to the input parameters *support* and *confidence*; both are values between 0 and 1 that are read from the table `Q (supp, conf)`. Q must contain 1 row, if you want to run your code with other parameters – update the values in Q. We will test your solution with different values of support and confidence.

```
julia_cs500> select * from Q;
supp | conf
-----+------
500  | 0.55
(1 row)
```

- We are only interested in CARs with one of two classification outcomes: *vdecile=1* (low violent recidivism score) or *vdecile=2* (high violent recidivism score). For example, we may compute the following rules:

```
age=30, gender=1 => vdecile=1
age=40, race=1 => vdecile=2
```

But we don't' compute the rules:

```
gender=0 vdecile=1 => age=30
age=30 => marriage=2
```

In other words, all rules we compute have *vdecile=1* or *vdecile=2* on the right hand side, and don't not have either *vdecile=1* or *vdecile=2* on the left hand side.

Your implementation should complete the code that is provided in the file Fairness.sql. Your solution should be entirely in that file, and should only use "regular" SQL queries: no procedural extensions, no loops. It is allowed to create additional relations. Do not change the schemas of the output relations.

You should not explicitly mention by name (hard-code) attributes other than *vdecile* and *race* anywhere in your SQL code. Assume that your code should still work if another regular attribute is added, e.g., income or education. It is OK to refer to *vdecile* and *race* by name, since these attributes have a special meaning.

Part 1 (50 points) We computed CARs with 1 attribute on the left and stored them in R2. Using the computation of R2 as an example (with intermediate relations C2 and F2), compute CARs with 2 attributes on the left, and store them in R3. The schema of R3 is given, do not change the schema. Important: make sure that attribute names in the body of the rule are listed in lexicographic order (i.e., attr1=age and attr2=gender is a valid rule, while attr1=gender and attr2=age is not).

Part 2 (25 points) Compute the impact ratio of CARs of the form X => C, where X is a regular attribute (not *race* or *vdecile*) and C is the positive outcome (*vdecile=1*). Impact ratio is the ratio of positive outcomes for the protected group S+ (*race=1* in our case) over the general group S- (race=2 in our case). Use the formula:

ImpactRatio (X => C) = support(S+ X => C) / support (S- X => C)

Only compute impact ratio for rules for which X => C, S+ X => C and S- X => C all pass the support and confidence thresholds. Write one or several SQL queries that compute ImpactRatio(X=>C) for all such rules. Store your result in IR. The schema of IR is given, do not change the schema.

Part 3 (25 points) Compute the elift ratio of CARs of the form S+ X => C, where S+ is the protected group (*race=1*), X is a regular attribute (not *race* or *vdecile*) and C is the positive outcome (*vdecile=1*). Use the formula:

EliftRatio (S+ X => C) = support(S+ X => C) / support (X => C)

Correct output for 2 cases is given in

https://www.cs.drexel.edu/~julia/cs500/documents/code/hw6_out1.sql
https://www.cs.drexel.edu/~julia/cs500/documents/code/hw6_out2.sql