Project: Data Mining with Declarative Programming

CS 240: Databases and Knowledge Bases, Spring 2018
Instructor: Prof. Carlo Zaniolo
Due on: Friday, June 15, 2018, 11:59pm

1 Project Objective

The goal of this project is to implement some popular data mining algorithms in declarative systems and explore the challenges and difficulties faced from a programmer's point of view. You will need to write a report commenting on the following:

1. difficulties you faced in implementing these algorithms and
2. ease-of-programming in these declarative systems.

2 Declarative Systems

You will implement the data mining algorithms on two kind of systems:

1. *Datalog system*: Use DeALS for this.
2. *RDBMS*: You can use the IBM DB2 Express Edition[1] for this. You are also free to use any other SQL system like MySQL, but some SQL systems do not support recursive queries which later on could become a hindrance.

3 Data Mining Algorithms

3.1 Naive Bayes Classifier (NBC)

The Naive Bayes Classifier[2] is one of the most simplest, yet popular machine learning model due to its interpretability and ease of use. Your task will be to write a **general purpose** NBC in SQL. Please note, a general purpose classifier should work on any structured dataset, irrespective of the number or data type of the columns and the number of classes. You do not have to implement this in DeALS. A simple NBC version implemented in DeALS is available for reference[3].

[1] `https://www.ibm.com/developerworks/downloads/im/db2express/index.html`
[2] `https://en.wikipedia.org/wiki/Naive_Bayes_classifier`
[3] `http://wis.cs.ucla.edu/deals/tutorial/nbc.php`

3.1.1 Things to consider

1. How do you handle numeric attributes – (i) applying discretization or (ii) using Gaussian distribution.

2. How do you handle missing values – (i) replace with median/mode, (ii) ignore them or (iii) treat missing value as an attribute value.

3. How to make your classifier work on any number of columns – (i) one option is to represent the data in a vertical format[3] i.e. a horizontal row in the format `<ID, ColumnA, ColumnB, ...>` is expanded into multiple rows as in `<ID, ColumnA, ValueForA>, <ID, ColumnB, ValueForB>`

4. How do you handle frequency zero problem i.e. seeing a combination during prediction which you have not seen during training – (i) One option is to apply Laplace smoothing technique (for simplicity you can just use add-one smoothing).

3.1.2 Datasets

Test your classifiers on the following datasets:

1. *Mushroom Data Set*[4]: Contains all categorical attributes with many missing values.

2. *Bank Marketing Data Set*[5]: Contains categorical and numeric attributes. Numeric attributes can be discreet as well as continuous real.

3.1.3 SQL Queries

Your SQL queries should perform the following:

1. Load the data into a table.

2. Partition the data in table into training and test data sets in the ratio of 4:1.

3. Verticalize the training table.

4. Build a simple NBC model on the verticalized format. You may need additional queries for handling numeric attributes and missing values. The model parameters can be stored in another table.

5. Using the model parameters, predict the accuracy on the test table.

3.2 K-Nearest Neighbors Classifier (KNN)

K-Nearest Neighbors classifier[6] is a non-parametric machine learning classification model that predicts the class of a test instance based on the class labels k training instances closest to it. Your task will be to write the K-NN classifier in SQL and in DeALS.

[4] `https://archive.ics.uci.edu/ml/datasets/mushroom`
[5] `https://archive.ics.uci.edu/ml/datasets/bank+marketing`
[6] `https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm`

3.2.1 Things to consider

1. What distance metric to calculate the distance between two instances – (i) One option is to use Euclidean distance.
2. Can this query be implemented without recursion?
3. If implemented with recursion, can `PREM` make this computation more efficient during nearest-K calculation?
4. Can you justify your answer with a theoretical analysis?

3.2.2 Dataset

You can test both the systems on *Hill-Valley* data set [7].

4 Extra Credit

Apart from the algorithms mentioned in Section 3, pick any data mining algorithm of your choice and implement it in SQL and Datalog. Can you make your algorithm more efficient by applying *pre-mappability* `(PREM)`? If so, justify.

One option is to look into the PageRank[8] algorithm and try to implement it on a very small subset of citation networks[9] data.

5 Submission Instructions

- You need to submit a report, a README file, the code and the datasets (only those datasets, where some pre-processing was done) as a zip folder on CCLE.
- **The README file should contain complete instructions on how to run and test your code.** To elaborate, the README should clearly mention what relational tables need to be created, what should the schema for the tables be, what SQL system needs to be installed, which queries need to be executed and in what order. It is better to write a setup script that does these aforementioned tasks when executed.
- If you are not able to include your pre-processed datasets in the zip package, please make sure to include a pre-processing script that would download the corresponding dataset(s) from the internet and process it.
- Your report should be in PDF.
- Please mention full names of all the team members, their UCLA IDs and email addresses on the first page of the report.

[7] `http://archive.ics.uci.edu/ml/datasets/hill-valley`
[8] `https://en.wikipedia.org/wiki/PageRank`
[9] `https://snap.stanford.edu/data/cit-HepPh.html`