

REGRESSION TREE MINER

Index:

1. Introduction
2. Installation
3. Material
4. User guide
5. Test cases

Introduction

“Regression tree miner” is a client/server application which incorporates data mining features, a branch of machine learning and statistics used to discover patterns in large data sets, to allows users to predict the value of a continuous attribute associated with a vector of independent attributes.

Study case developed by Fabio Fucci for the course advanced programming methods held by the university of Bari Aldo Moro.

Installation The user will be provided with three .bat files (one for the server, one for the client CLI and one for the client GUI) and with a series of sql scripts. For the application to properly work the computer which will run the server side application must have a working version of mysql installed (at least version 8.0.19) with database MapDB, which will be used to store the various tables, and the user MapUser with password “map”, which will be used to access the database. The scripts in the SQLScripts folder can be used to create the database, the user and insert in the database all the tables used as test cases, the only difference between the file noDropAndCreate.sql and dropAndCreate.sql is that the second one will remove the database with all the existing tables if it exist, so use with care. To run the server application double click on the file mapServer.bat, a command line interface will open waiting for connections. Each time a client connects the server will display it's id and the current phase.

To run the client application double click either on MapClientCLI.bat or MapClientGUI.bat, further explanation in the user guide section.

User guide

The CLI version of the client can be executed by double clicking on the file mapClientCLI.bat, by doing so it will automatically connect on localhost port 8080, if the server is hosted somewhere else the user can either modify the bat file or open the terminal and input “java -jar path/to/client/jar server_ip server_port”.

Instead on the GUI version these settings can be modified visually once opened, as usual, by double clicking on the file MapClientGUI.bat.

In both version of the app the user can choose to learn a new regression tree from the database or to load it from file, In the first case after entering the table name the tree will be learned from the data set and later saved to a file with name table_name.dmp, in the second case the user must be careful to only enter the file name without any extension and the tree will be loaded to memory.

In GUI version of the software, the user can modify all the setting before pressing the connect button, if there are problems connecting to the server or retrieving the table a message will be displayed in the console, else if the connection is successful the user can start the prediction phase by pressing the predict button.

Test cases

All the test cases shown can be reproduced by entering the table name of the specific test.

Only one test is shown with the GUI since the identical to the CLI.

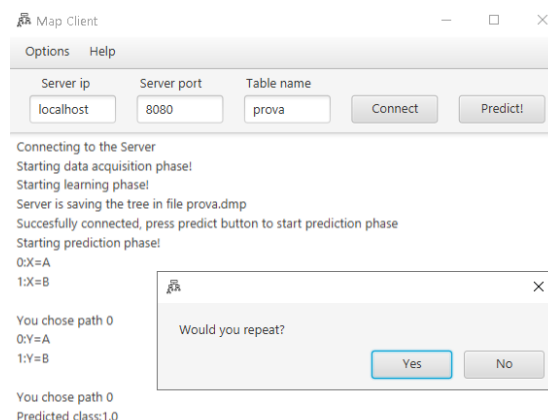
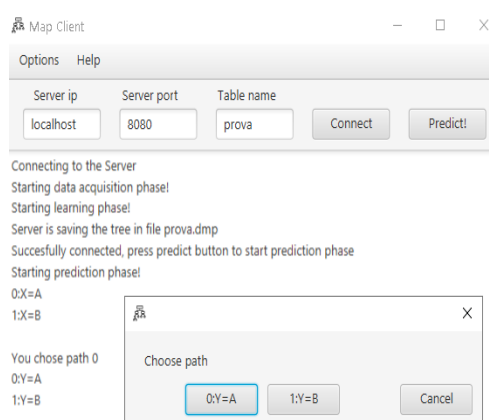
- prova, small test case table with two independent discrete attributes.

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Fabio\Desktop\Progetto>java -jar jars/mapClientCLI.jar localhost 8080
Socket[addr=localhost/127.0.0.1,port=8080,localport=9481]
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
Table name:
prova
Starting data acquisition phase!
Starting learning phase!
Starting prediction phase!

0:X=A
1:X=B

0
0:Y=A
1:Y=B

0
Predicted class:1.0
Would you repeat ? (y/n)
n
```



- provaC, changes one of the attributes of prova to a continuous attribute.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\Fabio\Desktop\Progetto>java -jar jars/mapClientCLI.jar localhost 8080
Socket[addr=localhost/127.0.0.1,port=8080,localport=9510]
Welcome! press 1 to learn from data, press 2 to load from file!
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
Table name:
provaC
Starting data acquisition phase!
Starting learning phase!
Starting prediction phase!

0:X=A
1:X=B

0
0:Y<=2.0
1:Y>2.0

0
Predicted class:1.0
Would you repeat ? (y/n)
n
```

- servo, biggest test table with four independent discrete attributes.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\Fabio\Desktop\Progetto>java -jar jars/mapClientCLI.jar localhost 8080
Socket[addr=localhost/127.0.0.1,port=8080,localport=9512]
Welcome! press 1 to learn from data, press 2 to load from file!
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
Table name:
servo
Starting data acquisition phase!
Starting learning phase!
Starting prediction phase!

0:pgain=3
1:pgain=4
2:pgain=5
3:pgain=6

0
0:motor=A
1:motor=B
2:motor=C
3:motor=D
4:motor=E

0
Predicted class:4.588888888888889
Would you repeat ? (y/n)
n
```

- servoC, changes two of the attributes of servo to a continuous attribute.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\Fabio\Desktop\Progetto>java -jar jars/mapClientCLI.jar localhost 8080
Socket[addr=localhost/127.0.0.1,port=8080,localport=9514]
Welcome! press 1 to learn from data, press 2 to load from file!
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
Table name:
servoC
Starting data acquisition phase!
Starting learning phase!
Starting prediction phase!

0:pgain<=3.0
1:pgain>3.0

0
Predicted class:3.2061224489795923
Would you repeat ? (y/n)
n
```

- allIdenticalValues, data set used to show the case in which the decision tree only has one leaf node.

```

C:\WINDOWS\system32\cmd.exe

C:\Users\Fabio\Desktop\Progetto>java -jar jars/mapClientCLI.jar localhost 8080
Socket[addr=localhost/127.0.0.1,port=8080,localport=9765]
Welcome! press 1 to learn from data, press 2 to load from file!
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
Table name:
allIdenticalValues
Starting data acquisition phase!
Starting learning phase!
Starting prediction phase!

Predicted class:4.392857142857143
Would you repeat ? (y/n)

```

- allIdenticalValuesC, like the previous one but with continuous attributes.

```

C:\WINDOWS\system32\cmd.exe

C:\Users\Fabio\Desktop\Progetto>java -jar jars/mapClientCLI.jar localhost 8080
Socket[addr=localhost/127.0.0.1,port=8080,localport=9767]
Welcome! press 1 to learn from data, press 2 to load from file!
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
Table name:
allIdenticalValuesC
Starting data acquisition phase!
Starting learning phase!
Starting prediction phase!

Predicted class:4.392857142857143
Would you repeat ? (y/n)
n

```

- emptyTable, used to show what happens if the table is not populated.

```

C:\WINDOWS\system32\cmd.exe

C:\Users\Fabio\Desktop\Progetto>java -jar jars/mapClientCLI.jar localhost 8080
Socket[addr=localhost/127.0.0.1,port=8080,localport=9769]
Welcome! press 1 to learn from data, press 2 to load from file!
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
Table name:
emptyTable
Starting data acquisition phase!
There are no examples in this table

```

- oneColumn, used to show what happens if the table only has one column.

```

C:\WINDOWS\system32\cmd.exe

C:\Users\Fabio\Desktop\Progetto>java -jar jars/mapClientCLI.jar localhost 8080
Socket[addr=localhost/127.0.0.1,port=8080,localport=9771]
Welcome! press 1 to learn from data, press 2 to load from file!
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
Table name:
oneColumn
Starting data acquisition phase!
The table has less than two columns

```

- nullTuple, used to show what happens if one or more tuple in the table has null values.

```

C:\WINDOWS\system32\cmd.exe

C:\Users\Fabio\Desktop\Progetto>java -jar jars/mapClientCLI.jar localhost 8080
Socket[addr=localhost/127.0.0.1,port=8080,localport=9773]
Welcome! press 1 to learn from data, press 2 to load from file!
Learn Regression Tree from data [1]
Load Regression Tree from archive [2]
1
Table name:
nullTuple
Starting data acquisition phase!
The table contains a null tuple

```

Material

- mapServer.bat and mapServer.jar.
- mapClientCLI.bat and mapClientCLI.jar.
- mapClientGUI.bat and mapClientGUI.jar.
- SQL scripts to create the database and the various test tables.
- Source code.
- UML class diagrams.
- Jar file for JDBC driver connector version 8.0.17.
- Public javadoc for the users.
- Private javadoc for the developers.