Predicting Automobiles' prices

User Manual

Case Based Reasoning Approach

Advanced Machine Learning Techniques

Universitat Politècnica de Catalunya
Faculty of Informatics

23/01/2017

Team members

Kazancli, Erol

Kazimi, M. Bashir

Mahyou, Khalid

Sanz Bertran, Nil

Supervisor Prof. Sànchez i Marrè, Miquel

Subject MAI - AMLT

Table of Contents

1. CBR engine goals

As some researchers defined, CBR can be defined as follows: “transferring knowledge from past problem solving episodes to new problems that share significant aspects with corresponding past experience and using the transferred knowledge to construct solutions to new problems”. Therefore, we can define CBR as a methodology of solving new problems by adapting the solutions of previous similar problems. In other words, use past experience in a new problem. CBR proves to be quite useful when the domains are difficult to formalize, i.e. making a rule-based system is not feasible or manageable.

In this work, we have build a CBR system to predict car prices based on the data set of cars and their relative prices. This document comes along with the detailed report of the work and serves as a user manual for the system. The following sections are allocated for describing the Startup/Shutdown of the system, its functionalities, and usage example.

2. System Startup/Shutdown

The user must open a terminal window on a Linux System and change the directory to the folder which contains the source code and files submitted alongside this user manual. Once in the relevant directory, the system could start with the following command in terminal:

```
python CBR_test.py
```

The command reads data from two files in the current directory; *test.csv* containing the test set, and *train.csv* containing the training set. It uses the CBR model implemented in *cbr.py* file to retrieve *k* most similar instances for each test instance where *k* is fixed to be 5 in the code. It takes the average price of the *k* most instances to use as a prediction for the test instance. If the similarity is low, meaning the similarity of the test instance with the most similar training instance is higher than a threshold, which we set as *3*, we add it to our case base so that it could be used for predicting other similar cases.

Finally, to evaluate our system, we calculate the average error between the predictions for the test set and the real prices and print it out to terminal. We also save the final case base to a file named *train2.csv.*

3. System functionalities

The system has different parts. First, the system must be trained in order to get accurate results. For this purpose, we have a database with enough samples that the system loads when it first start running. When loading the dataset, the system might perform some preprocessing tasks if it is necessary.

On the other hand, users can ask the system introducing a car’s features. The system will preprocess this input together with the existing case base and will output a predicted price for that car.

4. Use example

The data set for this work has been retrieved from the UCI Machine Learning repository, as stated in the detailed report attached, but the system could be used for predicting car prices for any car dealership agency to set a price for a new car that they put on sale. Another use of this application could be for private individuals who want to buy a new car. They could input their desired features for a new car and estimate how much it would cost them and they could plan their budget accordingly. Finally, it could be also used if some individual wants to sell their car and need to set an initial reasonable price. They could use our system for an initial guess on how much it would sell for.