Predicting Automobiles' Prices

Technical Report

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. Introduction

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.

1.1. CBR principles

The CBR process can be broken down into the following steps (as shown in **Fig. 1**):

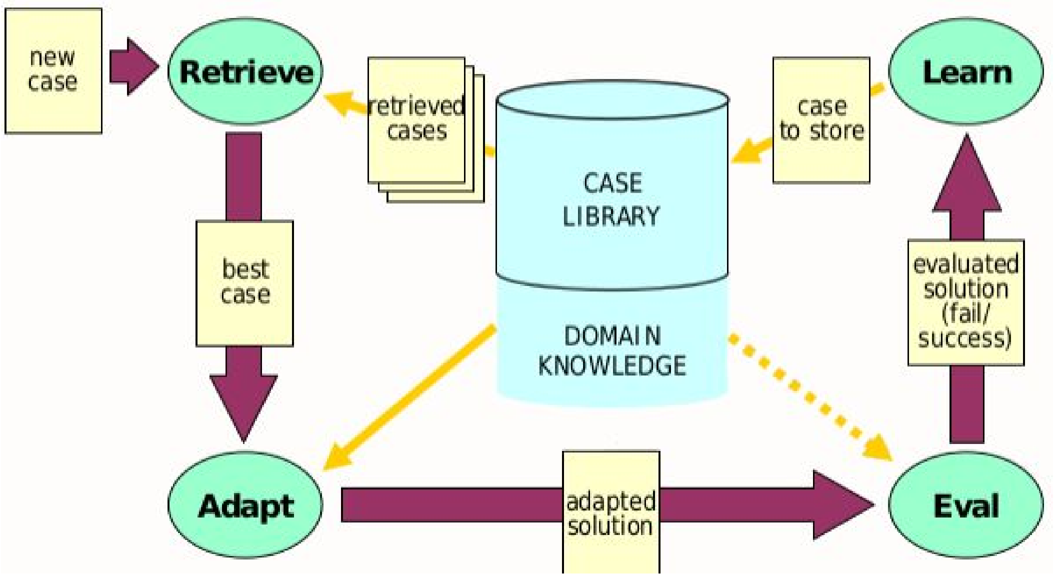

***Fig. 1** CBR cycle*

1. **Case and case library representation**: The content and the structure of a case is defined.
2. **Retrieve**: Search and get from the case library the similar (or most similar) case/s to the new case.
3. **Reuse (Adaptation)**: The knowledge in the retrieved cases is used to obtain a solution.
4. **Revision (Evaluation)**: The solution obtained is revised to adapt it better to the new case.
5. **Retention (Learning)**: The part or all the knowledge obtained from the new case is stored in the case library.

1.2. Chosen application domain

One of the big challenge for automobile sector is how to predict the price for a new car given several variables. They may be interested in knowing a new car's price for different reasons: (1) A company has created a new car and wants to know which should be its price based on features that share with other vehicles they already know the price. (2) Another reason could be that a company is interested to know the price for a competitor's new car and act in consequence. (3) One other reason, and important one, is to know the price of a used car. For instance, a customer could use a system that given several features of a car, the system is able to give a value for that car.

In this project, we wanted to solve the problem using the CBR approach with its different components and see how it performs in predicting the car's price given several features of that car.

2. Requirement Analysis

In this section, we present the requirement analysis for our CBR engine to predict automobile prices. First, we research about the context analyses, focusing on similar systems. Then, we define our target sector and the specific purpose of the system. Finally, we present what are the main characteristics of the system.

2.1. Context Analysis

For this task, we wanted to develop a project based on case based reasoning (CBR) methodology. At the very beginning we had in mind different domains where to apply CBR and after some discussions and research, we ended up choosing to apply this methodology to a very interesting domain, which is to predict automobiles' prices.

Surprisingly, by doing some research we have found that work on estimating the price of cars is very recent yet very sparse. Most of the techniques and methodologies applied to this domain, are related with machine learning. For instance, [1] showed that using support vector machines (SVM) with genetic algorithms to find the optimal parameters for SVM give a high accuracy and very good results. In another work [2]. the authors used multiple regression analysis to find the optimal automobile's price, also given very high accuracy.

In this project, we have implemented a CBR to predict the automobiles' price and be able to give a value for a particular car with high confidence.

2.2. Purpose of the system

As we mentioned in the previous section, there are different ways where we can apply this system. It can be applied by an automobile company to predict which will be the price for its new car based on past cars with similar features. It can be used, also, by an automobile company to predict the price of their competitor's new cars' price. And the last option we thought about is to use the system to predict the price for an used car. In this case, a user instead of buying a new car, he/she can wait for a short period of time and use the system to predict which will be the price of his/her interested car based on several features.

In this project, we aim to help people to buy an used car by developing a case base reasoning system and used it to predict a car's price based on several features of that car.

3. Functional Architecture

We have selected a dataset from the UCI repository related with automobiles [3]. The dataset contains 205 instances, with 26 attributes. The attributes can contain missing values. The list of attributes can be found in the dataset website. Both categorical and numerical features are present in the dataset. These are some of the attributes that we consider relevant for our assignment:

- Make: The brand of the car (audi, bmw, …)
- Fuel type: Gas or diesel
- Aspiration: std or turbo
- Number of doors: two or four
- Engine location: front or rear
- Curb-weight: Continuous from 1488 to 4066
- Num of cylinders: 8, 4, 6, …
- Engine size: Range [64, 326]
- Horsepower: Power of the engine, range [48, 288]
- City mpg: Average consumption in city
- Highway mpg: Average consumption in highway
- Price: Price of the car in range [5118, 45400]

We will use the “price” as the single output of the dataset, and the rest of the features as inputs.

The dataset has been divided into a training set of 146 instances and a test set of 51 instances.

The dataset contains categorical values encoded as strings that must be converted to numerical values in order for the system to work. The missing values have to be controlled as well.

The application will preprocess the data obtained from the CSV files. The missing values of the fields are replaced for the mean value in the numerical features. In the categorical features, the missing values are not replaced but they are handled as another class or category in that feature.

The final model will be a regression model that will make use of the training case base to predict the final price of the car. It is therefore a instance based learning model where the training set is used each time to produce a prediction [4]. There is no training involved offline to create a model.

The model works in a similar way as a k-nearest neighbours algorithm. The query instance where we would like to predict its price is compared against the instances from the training set. A similarity measure is calculated for each of the training instances against the query instance.

This similarity measure takes into account all the input features of the training set both numerical and categorical. For each instance, a difference of values is calculated. For the categorical values, the difference is simply 1 or 0 depending if the category matches the query and the training instance for that categorical feature. All the differences are summed together to get the final similarity. In this case it's a distance metric where a larger number represents less similarity.

On the other hand, some weights have been added in each feature in order to improve the similarity metric used. The final distance value is obtained by the scalar product (inner product) of the weights vector with the vector of differences. The weights have been defined manually to obtain better prediction performance on the training set.

4. Proposed CBR engine

As we can see in **Fig. 2**, the input of our system is a set of features describing a car (depicted in the figure by a blue circle) and the output is the predicted price (depicted by a blue circle in the figure). Between the input and the output, we have four different steps. We start with preprocessing all the input data. As described in the previous section, the preprocessing step take care about missing values and takes care about the categorical one.

The next step is to retrieve the k-most similar cases from the case library to the input case, as described in the previous section, and then reuse these retrieved cases to adapt them, if it is necessary, to the input case and propose a price. Last, but not least, we store to the case library the new input case, if there is no other similar cases (if the minimum distance to the samples in the case base is bigger than a threshold value).

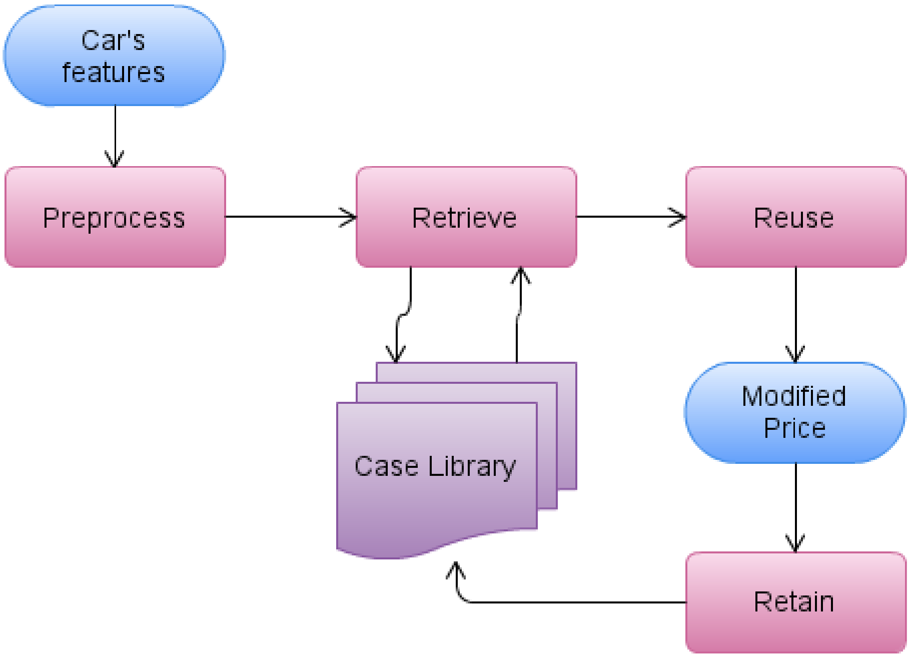

***Fig. 2** CBR solution design*

4.1. Case Structure and Case Library structure

Case Structure

We represent a case as a list of features describing a car. The following is a list of features we consider in our case library. There are 25 features describing a car plus the last attribute in the list that is the price.

*(symboling, normalized-losses, make, fuel-type, aspiration, num-of-doors, body-style, Drive-wheels, engine-location, wheel-base, length, width, height, curb-weight, engine-type, Num-of-cylinders, engine-size, fuel-system, bore, stroke, compression-ratio, horsepower, peak-rpm, city-mpg, highway-mpg, **price**)*

For instance a case stored in the case library might look like:

(*0, 81, toyota, gas, std, four, wagon, 4wd, front, 95.70, 169.70, 63.60, 59.10, 2290, ohc, four, 92, 2bbl, 3.05, 3.03, 9.00, 62, 4800, 27, 32*, ***7898***)

On the other hand, an input to the system, might look like the previous list but without the final component, i.e. the price, because this will be predicted by the system.

Case Library Structure

Our library is a collections of cases, as described in the previous sub-section, forming an attribute-value vector. The attribute is a list car's feature and the value is the final price of the car for those attributes.

4.2. CBR cycle implementation

This section describes the implementation of each of the CBR cycle step.

Retrieval

In this step, we are using the *k-nearest neighbours* to retrieve the most similar cases to the input case.We tested several k values for the number of nearest neighbours to calculate the price from and found that k=13 was the ideal number of nearest neighbours in this case to get the lowest average error with the test set.

	k=3	k=8	k=13	k=18	k=20
Avg. error	2066	1603	1551	1590	1597

The similarity is measured by taking weighted differences for each feature and adding them up. We observed that some features such as curb-weight and horsepower are more important in predicting prices, and, therefore, we gave such features more weight in similarity calculation. Of course, it would be much better to learn these weights with a training data beforehand.

Adaptation

After having collected the most similar cases in the previous step, in this one we take the *weighted average* of the most similar cases to predict the input case price. By weighted average we mean proportional to the inverse of distance to the similar case. Here, beware that the weights are different than the ones used in similarity calculation.

Evaluation

We do not apply any revision to the solution obtained, since we do not have any benchmark information to evaluate the quality of the prediction we make. If we had certain general rules, such as the price range for a given make, than we would check the prediction we made with this information and we could reject or accept accordingly.

Learning

In this last step of the CBR cycle step, we add the our case library the new input case only if the minimum distance from this new case to the k-most similar, retrieved in the first step, are bigger than an *alpha* value. In our particular case, we set *alpha = 3*.

5. Proposal of real application

We have seen how the CBR system can be used to predict prices of cars based on the attributes of each car.

Nowadays, lots of cars are sold in second hand because they are much cheaper than the new ones. The fact that cars nowadays have much less problems than in the past and live longer is one of the reasons why people are more confident in buying second hand cars. The sell of second hand cars is increasing in Spain, that reached more than 1.8 million sold cars in 2016 [5].

One of the most used channels to buy and sell second hand cars are the websites and platforms available in the Internet. The play an important role in this market, allowing the sellers to publish the cars with all the description, features and photos, and buyers to find the desired car and finally contacting the seller.

The CBR system implemented could be used by this platforms to automatically adjust the price of the cars based on the attributes and the market supply and demand.

5.1. Functionality

Usually, the sellers first publish the car with a high price. Then they manually reduce the price periodically if the car is not sold in order to sell the car as expensive as possible.

Based on the car’s features, and the current market, the platform will adapt the price of the car dynamically in order to maximise the probability of selling it for a good price.

The platform would have to request as much information from the car as possible. That could be provided by the seller of the car, or using a more user experience friendly interface where the seller specifies the brand, model and year of the car, and the system automatically recognises the different features and allows the seller to select the dynamic features that the car might have.

Once all the attributes of the car are obtained and the car is published in the platform, then the system will update the price periodically. The owner can still specify the minimum price that it can reach, or the update periodicity of the price.

5.2. Implementation details

The system would have for each published car, a list of attributes. This list of cars are the case base of the system that will be used as a training set to predict the prices.
The feature representation of the cars will contain the attributes related to the car. It will also contain other important features:

- Year: Year of the car
- Mileage: Total distance traveled or covered by the car.
- Location: Physical location of the car. Could be represented in geographical regions like counties or provinces. It could also be represented as the geographical coordinates.

The details of the CBR system are explained in the following sections.

Retrieve

The case based consists in a list of cars and a final sold price. The sold price is obtained from real car purchases performed in the platform. In the retain phase, new instances will be created.

The system periodically updates the price of the cars. It retrieves from the case base the most similar cars based on the features. The categorical values are also used in the similarity calculation. As the features are not represented as numerical vectors, it is necessary to perform a similarity calculation for each of the instances in the case base. It is important then to keep a relatively small amount of training instances since no indexing trivial is applicable based on K-D Trees. The retain strategy will also ensure a limited amount of training instances.

Reuse

The price of the car would be the weighted average of the prices from the retrieved instances.

Revise

The revise strategy evaluates the quality of the price prediction. If the price recommendation is incorrect, then either there won't be any buyer interested in buying the car (expensive price), or there will be too many buyers that are interested (cheap price). Based on the regions where the car is sold, the system can obtain the average number of buyers that are interested in a car before it is sold, and the period of time necessary to sell the car. Both values are used to evaluate the prediction quality.

After a specific period of time, the evaluation is performed. If the number of buyers that have contacted the seller matches with the average behaviour for sold cars, then the recommendation is marked as correct, and the published price will not change. In the contrary case, the recommendation is marked as incorrect, and the price is adjusted based on the number of buyers and period of time. The more the interested buyers, the higher the price will become.

Retain

The retain strategy will focus on two things: add new instances to the case base (retain strategy), and delete useless instances from the case base (forgetting strategy). The forgetting strategy is important to keep a limited size in the training set in order to allow the nearest neighbour search from the retrieve phase to be computed using a brute force search.

Once a car is marked as sold, the system will use the sold price to add the new instance in the case base. Only those prices that are marked as correct will be added, so that the system will not add incorrect predictions even though the car has been sold.

There will be two different types of forgetting strategy.

The first one will be based in the date of the purchase. The case base will not contain cars that have been sold long time ago. The system will periodically delete instances that represent purchases with a specific date or older. That date is calculated by applying a difference to the current date. This forgetting strategy will ensure that prices are not based on market situations from long ago that are no longer applicable in the present.

The other forgetting strategy will be based on density in the case base. It will ensure a limited density in all the regions of the training space. Using a shuffled ordered training set, for each training instance, the system will retrieve all the instances that that are within a specific similarity. If there are a lot of neighbour instances then the query instance will be deleted from the training set. This will delete useless instances since most of the instances in that specific region of space already contain the necessary information to create the price prediction.

6. Conclusion

We admit that CBR is not the best option to predict automobile prices. However, even with the coarsely tuned system of ours, we got quite close results to the real price in many cases. This CBR system we propose may be improved by learning the weights, since the weights we proposed were only based on observation and trials. We observed that some features, such as curb-weight and horsepower, were so important that we got better results as we increased the weight for these features. This does not mean the other features are not important, they would rather play a role in better tuning the predicted price. However, it was not possible to try enough number of weight combinations, let alone all the combinations, with this many features. Therefore, an initial training step, where all these weights are learned, would provide a substantial improvement.

Other improvement would come from keeping a benchmark information to check the quality of the prediction and accept/reject accordingly. A rule-based evaluation step, a human expert checking the final result or a system based on the demand with the proposed price would increase the quality of the predictions and the health of the case base.

Also, instead of taking a fixed number of nearest neighbours, a distance condition could be imposed, meaning that only the nearest neighbours that are similar enough would be considered in predicting the price. This might also improve the quality of the predictions.

Bibliography

[1] Listiani M. Support vector regression analysis for price prediction in a car leasing application (Doctoral dissertation, University of Hamburg).

[2] Richardson MS. Determinants of Used Car Resale Value (Doctoral dissertation, The Colorado College).

[3] Automobile dataset. https://archive.ics.uci.edu/ml/datasets/Automobile

[4] Instance based learning. https://en.wikipedia.org/wiki/Instance-based_learning

[5] Second hand cars in Spain, https://www.motor.es/noticias/ventas-coches-ocasion-primer-trimestre-2016-2016 27229.html