Application of Data-Mining and Knowledge Discovery in Automotive Data Engineering

Jörg Keller[1], Valerij Bauer[1], and Wojciech Kwedlo[2]

[1] DaimlerChrysler AG, Machine Learning, FT3/KL, P.O.-Box 2360, D-89013 Ulm Germany
{joerg.keller, valerij.bauer}@.daimlerchrysler.com
[2] Technical University of Bialystok, Wiejska 45a, 15-351 Bialystok, Poland
wkwedlo@ii.pb.bialystok.pl

Abstract. In this paper the authors present a powerful and efficient alternative to Neural Networks (NN) by application of Knowledge Discovery and Data-Mining (KDD) methods for real world data in vehicle design, particularly for automotive Data Engineering (DE) mechanisms and processes. Typical tasks in automotive engineering are dependency analysis, classification of concepts and prediction of characteristic design parameters. From the point of view of a design engineer the main drawback of a NN-based approach is a lack of clear interpretation of the results. For classical, statistical tasks an application of an instance-based method, e.g. K-Nearest-Neighbors (KNN), represents an appropriate alternative for the engineer. By application of rule-based methods the authors demonstrate an alternate in conceptual design, which, in contrast to NN, allows to interpret the results and proof or enhance designers knowledge. The approach of this paper is based on a novel application of an Evolutionary Decision Rule Learner with Multivariate Discretization (EDRL-MD) for classification, and of M6 for regression learning.

Keywords: Automotive Data Engineering, Evolutionary Decision Rule Learning, Knowledge Discovery, Data-Mining

Introduction

In automotive engineering designers have the task to develop vehicle concepts, which are limited by governmental restrictions, e.g. exhaust emissions and fuel consumption, and in regard to the customers requests, e.g. power, torque, acceleration or maximum speed. Development cycles of vehicles decrease constantly. More product niches are served by the manufactures. Therefore new vehicle concepts have to be developed rapidly.

In motivation, to overcome the time consumption by Computational Aided Design (CAD), traditionally the application of NN was a favourite AI technique over the last decade. But the results of NN can not be visualised, it is more like a black-box for the design engineer, who wants to proof the results for plausibility. Rule-based systems would offer a well-performing, alternative. Derived design rules are transparent and can be duplicated by the design engineer. Due to data quality, non plausible results could be detected. For objective, machine learning (ML) methods will be used

D.A. Zighed, J. Komorowski, and J. Zytkow (Eds.): PKDD 2000, LNAI 1910, pp. 464-469, 2000.

inautomotive data engineering for Knowledge Discovery and Data mining (KDD). Design rules and knowledge could be extracted from the voluminous databases in automotive industry. In this paper we describe the ML method employment in automotive Data Engineering. In our approach we demonstrate, with M6 for regression learning, and a novel application of an Evolutionary Decision Rule Learner with Multivariate Discretization (EDRL-MD) for classification, a powerful and efficient alternative to NN for concept design in CAD.

In section 2 we will illustrate the used domain data (real world data), a vehicle data warehouse. Section 3 describes our applied methods of regression and classification in automotive Data Engineering. In section 4 experiments and the empirical results are demonstrated. In section 5 we will finish with our conclusions.

Automotive Data Engineering

ML methods and tools in a global product and tool developing process have to be evaluated for data engineering. Data fusion has to be done for an automated data engineering data consistency has to be proved constantly. Furthermore data has to be free of redundancy. Rezende et al. [3] describe how to generate a unified database interface for multiple heterogeneous databases for automotive application, which has been used by the authors as a data base for data-mining.

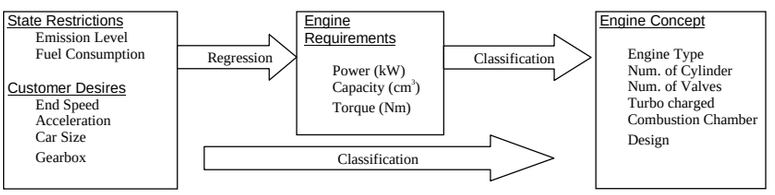

Fig. 1. Typical employment of classification and regression (prediction) in automotive data engineering

The data used for our analysis was taken from DaimlerChrysler internal databases. In our approach, ML for automotive data engineering, we have to employ with several layers of data: cylinder head, engine, drive assembly, car body, additional aggregates and or units. The engine, as subset, contains geometrical, mechanical or fluid data. Emission and fuel consumption are dependent on, e.g. turbulence generation in the combustion chambers, mixing of fuel injection and air. For our approach on ML in data engineering we selected core experiments, exemplary, to examine the subjects:

- vehicle-concept classification
- regression learning on exhaust emission and fuel consumption

A table was selected with more than 1036 records and 5 attributes, fuel consumption, exhaust emission, power, torque, max speed, vehicle weight, valve concept (no. of valves per cylinder) and gearbox type.

Employed Methods for Data Engineering

3.1 Employed Methods for Regression

In the table below the variables we took for our regression task are shown. Because of very high positive correlation between engine power engine torque of about $r \cong 0.99$, we agree on prediction only of engine power.

Table 1. Input and Output Variables for Regression

Input Variables	Var. Type	Output Variables	Var. Type
Car Weight	Numeric	Engine Power	Numeric
Car Speed	Numeric	Engine Capacity	Numeric
Acceleration	Numeric		
Gearbox Type	Symbolic		

3.1.1 NN

For our core experiments the training method *Quick* was selected in CLEMENTINE, SPSS/ISL 1998 [1], because of the higher accuracy. Prevent overtraining button was switched on and the training was stopped on default. 50% of the data set were chosen for training model and 50% for testing. The accuracy of the model was tested with cross-validation method by dividing the data set in 10 sub sets and using nine for training and one for testing for every sub set. The authors applied a NN capacity & power architecture for the experiments:

- Input layer: 4 neurones
- Hidden layer: 4 neurones
- Output layer: 2 neurones
- Predicted accuracy: 96.36%

3.1.2 KNN

The KNN Model for prediction of engine power and capacity were built on 3 Nearest Neighbours using the Euclidean metric for calculating distances. Six examples were removed from Data set to test them for prediction and compare with real Data. Results are shown in the Table 6.

3.1.3 M6

In our approach we used for regression modelling the M6 method of Quinlan [4]. By making regression with M6 only one output variable is possible to define at once. It was necessary to build one model for engine power and one for engine capacity prediction.

3.2 Employed Methods for Classification

For our classification experiments we used EDRL-MD system [2]. EDRL-MD learns decision rules using an *evolutionary algorithm*. The main novelty of this system consists in *multivariate discretization* i.e. the simultaneous discretization of all continuous-valued attributes.

The implementation of EDRL-MD used in our experiments employs a different fitness function than the original version described in [2]. The fitness of the ruleset RS_c the class c from the other classes is defined as:

$$f(RS_c) = \frac{\Pr(RS_c)}{Compl(RS_c)} \tag{1}$$

where $\Pr(RS_c)$ is the probability of a correct classification of an example given by:

$$\Pr(RS_c) = \frac{p + N - n}{P + N} \tag{2}$$

where the number of the positive and negative examples covered by the ruleset is denoted by p and n respectively; the total number of positive and negative examples in the learning set is denoted by P and N. $\mathrm{Compl}(RS_c)$ is the complexity of the ruleset given by:

$$\mathrm{Compl}(RS_c) = \alpha \log(L + 1) + 1 \tag{3}$$

where α is a user supplied parameter and L is the total number of elementary conditions (selectors) in the learning set. In our experiments we used $\alpha = 0.0005$.

Core Experiments and Test Cases

The test cases have been done for the parameters: engine power and capacity.

4.1 Regression in Automotive Data Engineering

Analysing the regression results, we noticed, that the highest deviation (Absolute Error>400 cm^3) by prediction of Engine Capacity occurred by high volume Engines (>3000 cm^3) with missing values. M6 maximum relative error for engine capacity prediction amounts 24% and maximum relative error for engine power prediction amounts 17%, those error rates are less than using NN method for regression. The correlation of relative errors for corresponding fields of NN-Model and M6-Model result in poor correlation (0.036 and –0.062).

4.2 Classification Learning

In this section the results of our classification experiment are presented. We compared the performance of EDRL-MD to that of two neural networks: multi-layer perceptron network (MLP) with one hidden layer and radial basis function network (RBFN).

Table 2. Results of KNN-Prediction for Engine Power and Capacity, REC Relative Error for Engine Capacity; REP Relative Error for Engine Power

	KNN-Cap.	KNN-Power	Capacity	Power	REC	REP
Vehicle 1	2330	100	2496	110	6,7(%)	9,1(%)
Vehicle 2	1995	79	1994	77	0,1(%)	3,0(%)
Vehicle 3	2381	102	1997	80	19,2(%)	27,1(%)
Vehicle 4	2496	110	2926	135	14,7(%)	18,5(%)
Vehicle 5	2524	107	3226	145	21,8(%)	26,4(%)
Vehicle 6	2151	92	2151	75	0,0(%)	22,7(%)

Table 3. Results of Prediction

Car Type	Capacity	M6-Capacity	Error(%)	NN-Capacity	Error(%)
Vehicle 1	2496	2571,1	3,01	2560	2,56
Vehicle 2	1994	1874,7	5,98	1803	9,58
Vehicle 3	1997	2076,5	3,98	2165	8,41
Vehicle 4	2926	3162,9	8,10	2787	4,75
Vehicle 5	3226	3489,5	8,17	3075	4,68
Vehicle 6	2151	2017,2	6,22	2180	1,35
Vehicle 1	110	108,3	1,51	112,0	1,82
Vehicle 2	77	59,2	23,13	61,0	20,78
Vehicle 3	80	79,9	0,18	80,0	0,00
Vehicle 4	135	132,6	1,79	136,0	0,74
Vehicle 5	145	156,9	8,20	146,0	0,69
Vehicle 6	75	76,2	1,63	76,0	1,33

Table 4. Classification errors and learning times in seconds for EDRL-MD, Multi-Layer-Perceptron (MLP) and Radial Basis Function Network (RBFN) .

METHOD	EDRL-MD	MLP	RBFN
ERROR RATE	25.2%	55.5%	34.5%
LEARNING TIME	313.2s	105.8s	59.9s

Both networks were trained using commercial CLEMENTINE [1] data mining system. The results of that comparison are shown on Table 4. The error rate estimated by ten-fold crossvalidation and the CPU time needed to build the classifier are presented. The learning time is the average of ten iterations of a single ten-fold crossvalidation run. All the algorithms were run on Sun Ultra-10 workstation with 300 MHz CPU. Using Mc Nemar's test we found that the difference in error rate between EDRL-MD and the other algorithms is highly (p-value<0.00001) significant.

Conclusions

This paper demonstrates that ML and DM give impact on a quick concept design description in automotive data engineering. In spite of the former used NN , rule-based methods offer for design engineers the advantage of comprehensible design rules, which are readable. The employment of the regression models for prediction of

engine requirements show, that both methods are able to make sufficing prediction from engineering point of view. For classification of design concepts EDRL-MD has the significantly lower error rate than MLP or RBFN. However this improvement is achieved at the expense of increased computational complexity. Nevertheless all methods can be successfully applied for engineering purposes, to help designer in the concept phase.

Acknowledgement

Wojciech Kwedlo was supported by Deutscher Akademischer Austauschdienst (DAAD) under the grant A/99/12865.

References

1. CLEMENTINE (1998). Data-Mining Tool, *SPSS/ISL, Reference Manual Version 5*
2. Kwedlo, W; Kretowski, M. (1999). An evolutionary algorithm using multivariate discretization for decision rule induction. Principles of Data Mining and Knowledge Discovery. *3rd European Symposium PKDD'99. Springer LNCS 1704.*
3. Rezende, F.d.F; Oliveira, G.d.S.; Pereira, R.C.G.; Hermsen, U.; Keller, J. (1999). A Unified Database Interface for Multiple Heterogeneous Databases. *Proceedings of 2nd Workshop on Engineering Federated Information Systems EFIS'99, May 5-7, 1999, Kühlungsborn, Germany, eds. Conrad, S.; Hasselbring, W.; Saake, G., infix*
4. Quinlan, J.R. (1992) Learning with Continuos Classes. *Basser Department of Computer Science, University of Sydney*