Using multivariate split analysis for an improved maintenance of automotive diagnosis functions

Jens Kohl*, Agnes Kotucz†, Johann Prenninger†, Ansgar Dorneich‡ and Stefan Meinzer§
*Validation & Verification System Functions and †Diagnosis After Sales
BMW Group
München, Germany
Email: {jens.kohl, agnes.kotucz, johann.prenninger}@bmw.de
‡Interactive Analyzer
Holzgerlingen, Germany
Email: dorneich@i-analyzer.de
§Pattern Recognition Lab
University of Erlangen-Nürnberg, Germany
Email: stefan.meinzer@faui51.informatik.uni-erlangen.de

***Abstract*—The amount of automotive software functions is continuously growing. With their interactions and dependencies increasing, the diagnosis' task of differencing between symptoms indicating a fault, the fault cause itself and uncorrelated data gets enormously difficult and complex. For instance, up to 40% of automotive software functions are contributable to diagnostic functions, resulting in approximately three million lines of diagnostic code. The diagnosis' complexity is additionally increased by legal requirements forcing automotive manufacturers maintaining the diagnosis of their cars for 15 years after the end of the car's series production. Clearly, maintaining these complex functions over such an extend time span is a difficult and tedious task. Since data from diagnosis incidents has been transferred back to the OEMs for some years, analysing this data with statistic techniques promises a huge facilitation of the diagnosis' maintenance. In this paper we use multivariate split-analysis to filter diagnosis data for symptoms having real impact on faults and their repair measures, thus detecting diagnosis functions which have to be updated as they contain irrelevant or erroneous observations and/or repair measurements. A key factor for performing an unbiased split analysis is to determine an ideally representative control data set for a given test data set showing some property whose influence is to be studied. In this paper, we present a performant algorithm for creating such a representative control data set out of a very large initial data collection. This approach facilitates the analysis and maintenance of diagnosis functions. It has been successfully evaluated on case studies and is part of BMW's continuous improvement process for automotive diagnosis.**

***Keywords*-Software maintenance; automotive diagnosis; multivariate analysis; split analysis**

I. Introduction

The amount of software functions and *electronic control units* (ECU) in automobiles has been steadily increasing since their first introduction in the 1970's. The autonomously operating ECUs from the early years of automotive electronics have subsequently been replaced by functions distributed over several ECUs collaborating over bus-systems, and in the last years by highly integrated components hosting dozens of functions. Both developments have led to a high complexity of the car's system architecture. For instance, modern luxury cars consist of up to 100 ECU hosting hundreds of functions communicating over five different bus-systems (cf. [1]). Recent studies estimate such cars having several millions lines of source-code [2]. With software functions taking over more and more (safety-relevant) functions in modern cars, their possible faulty behaviour has to be detected and dangerous effects prevented or mitigated. Additionally, information about the fault's root cause has to be provided to support repairs in the garage. These are the central tasks of automotive diagnosis. For the remainder of this paper we will make use of the terminology for errors, faults and failures as defined in [3].

II. Motivation

Legal requirements (e. g. §§133, 157 and 242 German BGB [4]) force OEM such as the BMW Group to support the after sales service of their cars and its components for 15 years after the end of the car's series production, effectively making diagnosis a significant economic factor in a car's life-cycle. For a detailed description of the car's life-cycle refer to [5].

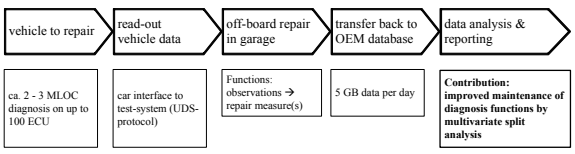

Fig. 1. Diagnosis incident and paper's contribution, based on [6]

Figure 1 shows a sequential overview of a diagnosis incident happening in the after sales service period as well as this paper's contribution. Diagnosis detects faults by taking observations and comparing the system's observed with its specified behaviour. In case of detected deviations, *diagnostic trouble codes* (DTC) [7] are stored in the ECU's *electrically erasable programmable read-only memory* (EEPROM) as symptoms to aid a later repair. Repair incidents for electronic components

1534-5351/11 $26.00 © 2011 IEEE
DOI 10.1109/CSMR.2011.42

in garages start with the Tester, the off-board diagnosis system, reading out the car's stored DTCs and on-board measurements. Additionally, the Tester hosts off-board diagnosis functions aiding the garage in vehicle repairs. Therefore the transmitted ECU data is analysed and complemented with additional observations such as reported customer complaints or garage off-board measurements. Based on these observations the Tester reasons for present faults and then suggests repair measures. Functional interconnectivity, hidden interactions or dependencies between on-board functions as well as propagating faults make the diagnosis' task of differencing between symptoms indicating a fault, the fault root cause itself and uncorrelated data increasingly difficult and complex. Selected data from repair incidents has been transferred back to the OEMs for several years now. Amongst the transferred data are diagnostic observations as well as repair measures. This data offers a very thorough insight into the cars' after sales behaviour for the OEM's quality departments, but with approximately 5 GB transmitted data **each** day (cf. [6]) very difficult to overview let alone analyse. Hence, without automatic support an efficient maintenance of thousands of diagnosis functions over such an extended life span -that almost equals the working life of an engineer- is very challenging.

III. Related Work

Several approaches have been used for statistical analysis in our problem domain. In this section we sum up related work partly done by other OEM. Buddhakulsomsiri and Zakarian [8] use data mining on warranty data but do not include on-board-diagnosis data. Blumenstock et al. [9] focus on on-board-diagnosis data to investigate specific failures occurring within a car while omitting warranty data. Both papers do not include off-board diagnosis. Müller et al. [10] merge diagnosis and warranty data, but do not create rules to detect repeat repairs. It aims to detect the garage's significant interactions to build rules which can then be distributed to all other garages. Sankavaram et al. [11] include all available data such as on- and off-board data as well as customer complaints. However, their use of unsupervised learning techniques obtains many faulty so called *spurious correlations*. In conclusion, the mentioned papers do not include all available data for their data basis and do not aim to detect insufficient diagnosis functions.

IV. Contribution

In this section we detail this paper's contribution. We propose a statistical approach improving the maintenance of diagnosis functions. We use *multivariate split analysis* on the data transmitted back to the OEM to automatically detect improvable off-board diagnosis functions, thus facilitating the diagnosis maintenance. A diagnosis function has to be maintained if its assigned repair measures, symptoms or related data are erroneous or insufficient. As mentioned before, with our domain's special characteristics such as hidden interactions and dependencies detecting such functions clearly is a tedious task. The contribution of this paper is an automatic detection of the significant symptom and fault pattern for a repair measure. Additionally, we introduce an algorithm filtering out all irrelevant data for a diagnosis function enhancing the split analysis' quality.

A. Discussing statistical analysis techniques

The choice of the statistical analysis method is strongly influenced by our domain's special characteristics. In this section we introduce these characteristics and discuss statistical approaches on their usability. With a huge amount of different variables which can be in relation to each other such as on- and off-board measurements as well as repair measures, we have to use **multivariate analysis** (cf. [12]). In a multivariate data analysis, a data subset is defined within a large data collection by restricting the value ranges of several data attributes to certain values or intervals, e.g. cars with defined failures or observation patterns occurring in a certain month. Although we seek relations between variables in the data, we focus on single variables instead of clusters, thus ruling out **cluster analysis** [13]. Clearly, the considerable data asks for an efficient analysis method which does not build redundancies, contradicting techniques such as **decision tree learning** [14] or **association analysis** [15]. Hypothesis verification by **split analysis** (test-control data analysis) [16] compares a 'test' data set having a certain property with a 'control' data set not displaying that property. χ^2-tests can be applied for verifying whether or not the test data significantly differs from the control data, for example comparing the average warranty cost per vehicle. Due to the huge amount of independent variables (more than 1.000 covariates) the problem of multicollinearity occurs. Because of that most of the above mentioned models are very unstable and yield to a very low rate of variance that is predicted. This also results from numerically instable compensation effects between almost collinear predictor variables: in the generated models, some variables have huge negative coefficients, other variables huge positive coefficients, and the net effect on the entire model output is almost zero. The split analysis method is able to deal with those problems very well so we use this technique to determine the related car data to repair measures.

B. Challenges applying the split analysis

We apply the split analysis to detect interrelations between technical vehicle information such as DTC and observable faulty car behaviour remedied with repair measures and recorded by textual repair findings. For each DTC we compare the car's repair events in which the DTC is found (test data) to the car's repair events in which this DTC is not found (control data). If the split analysis finds a significant correlation between the DTC's occurrences and repair measures, in particular multiple applications of specific repair measures, the DTC seems to be 'relevant'. If not, the DTC seems 'irrelevant' and should be removed from the diagnostic function. Problematic, however, when applying this approach is that test and control data can differ in an uncontrollable way due to many other *environmental variables* $E_1, \ldots, E_n$ (e.g. vehicle type or age, car extras, interactions with other functions,...). These variables might affect both the vehicle's data and observed

faulty behaviour. For instance, it might appear a certain DTC has an impact on a certain fault, but in fact -due to the car's functional connectivity- both of them have been caused by a third parameter such as an extremely low temperature, resulting in a non-causal relation between DTC and fault. Another example for this is diagnosis data from convertible cars liable to seasonal effects. Hence, the challenge is to filter out potential side-effects and cross-correlations which might have an impact on the analysis' 'target' property. Therefore, the challenge when applying split analysis method is to make the control data exactly representative for the test data with respect to all potential influence factors E_k which might be correlated with the occurrence of certain DTCs as well as the appearance of certain faults. In this case representative means that E_k's distribution is identical within test and control data.

C. Algorithm to create 'representative' control data subsets

As discussed above, the pre-existing control data is most likely not representative for the test data. With iteration step i, C_i denoting the control data after step i, N_i the amount of data records in C_i, $Distr_{k,i}$ the value distribution of variable E_k in the test data, $Distr_{k,0}$ being E_k's initial distribution, $diff$ as variable to measure the difference between two distributions of identical sets of values or classes (e.g. sum of squared differences of the values' relative occurence probabilities), Δ denoting $diff$'s maximum possible reduction obtainable by removing one data record from one of the distributions, N_i the number of data sets after optimisation step i has been applied and N_{stop} a user defined limit for the minimal amount of data records, we define an iterative algorithm to create a control data sample representative for the test data:

$\Delta = 0$
$diff = \sum_k (Distr_{k,0} - Distr_{k,T})$
while $((N_i >= N_{stop})$ && $(\Delta <= 0)$ &&
    ($diff$ stat. signif. > 0)){
    foreach (data record r_j in C_i){
      $C_{i+1,j} = C_i \setminus r_j$
      $Distr_{k,i+1,j} = E_k$'s value distribution in $C_{i+1,j}$
      $diff_{i+1,j} = \sum_k (Distr_{k,i+1,j} - Distr_{k,T})$
      $\Delta_j = diff_{i+1,j} - diff$
    }
    j_{best} = choose j for which Δ_j is minimal
    $\Delta = \Delta_{j_{best}}$
    $diff = diff_{i+1,j_{best}}$
    $C_{i+1} = C_{i+1,j_{best}}$
    i++
}

Alg. 1: Determine representable sample

In short, as long as we can reduce the overall difference of the environment variables' value distributions between test and control data by removing one single data record from the control data, we do so and remove the record which maximally reduces the difference. We stop if either the difference has fallen below a significance threshold or if the control data has reached a predefined size limit N_{stop}. One possible way for defining the criterion *diff is significantly > 0* is to use a χ^2-significance test [16]: $diff$ is significantly larger than 0 if a χ^2-significance test with the null-hypothesis *all E_k's value distributions are identical on the test and the control data* is rejected at a confidence level of more than 95%.
Conclusively, the algorithm removes data noise such as uncorrelated symptoms from the split sample.

D. Performance considerations

The basic algorithm described in the preceding section suffers from one serious disadvantage: its complexity is $O(N^2)$ with N the original number of data records in the control data. With the control data having several hundreds of millions data records in our domain, this basic algorithm does not deliver acceptable response times on currently available computer hardware. In order to overcome this problem, one of us designed a more sophisticated version of the above algorithm which is of complexity $O(N^{4/3})$. This improved algorithm was implemented into the module split analysis of the software *Interactive Analyzer* [17].

V. Evaluation

We evaluated our methodology on the automotive component *Adaptive Cruise Control* [18] with the Interactive Analyzer [17]. Table I shows run time measurements we performed with the Interactive Analyzer on a data set comprising vehicles whose internally recorded problem in form of DTC and warning history in form of on-board-measurements were read out each time a car came to the garage to fix a car's defect. The data set contains 6 columns *vehicle_readout_identifier*, *repair_cost*, *repair_date*, set-valued *DTC*, *vehicle_type*, *amount_previous_readouts*.

scale factor	1/4	1/2	**1**	4	8
data size in MB	57	113	**225**	900	1800
cars read out	39854	79708	**159416**	637664	$1.28*10^6$
DTC in million	1.23	2.46	**4.92**	19.7	39.4
runtime in secs	0.7	1.6	**3.2**	21	55
exp. runtime	0.5	1.3	**3.2**	20	51

TABLE I
Performance analysis algorithm 1

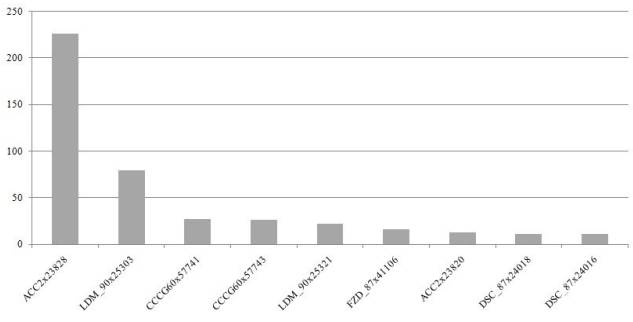

Fig. 2. Most frequent set DTC in case of repeated application of repair measure *exchange ACC*

Typically, between 30 and 200 recorded measurements can be found per car per read-out. As mentioned, the split analysis

aimed to detect whether one single DTC occurring in 30% of all readouts, has a significant impact on repairs and repair cost. The three environmental parameters e_1 for the car's *readout_date*, e_2 denoting the *vehicle_type* and e_3 the *amount previous readouts* serve to eliminate side-effects caused by differences in time/season/weather conditions, vehicle model type, and the overall vehicle quality (repair history). That means, the first step of the split analysis is to take a sample from the original control data of about 70% of the entire data. The sample should have exactly the same value distributions in the dimensions e_1, e_2, e_3 as the test data set of about 30% of the entire data. The original data base has a size of 225 MB and consists of 159416 readouts containing 4.92 million DTCs. We scaled our data sets ranging from $1/4$ to eight times its original size to measure our sampling algorithm's runtime and its scalability. We ran our algorithm on a Dell Inspiron 1525 notebook with 2GB RAM and Intel Core(TM)2 Duo 2.00GHz CPU.

Conclusively, our sampling algorithm's runtime behaviour approximates $O(n^{4/3})$. Additionally, it demonstrates the algorithm's capability enabling live analyses even on huge data with response times less than a minute, such as in our case study analysing the repair history of about 1 Million cars over 24 months.

Figure 2 shows an excerpt of an *Interactive Analyzer* analysis of the most frequent set DTCs when the repair measure *exchange ACC* was unsuccessfully taken and a repeat repair of the car occured. The DTCs are denoted with their number and by which ECU they were set. By analyzing the relations between set DTCs by different ECUs, we can determine a fault's propagation chain which is very useful when analysing functional interactions. In this case the analysis points out that *DTC 25303* which has the description "'LDM shut down because of ACC sensor"', set by the ECU *LDM_90*, which controls the vehicle's longitudinal dynamics, is clearly related to the ACC.

We analyzed and discussed the results of our methodology with the responsible diagnosis engineers and could validate about 80% of our results. Additionally, we were able to find new relations between DTC of different interacting components from diverse suppliers thus helping the OEM's diagnosis departments in analysing propagating faults.

VI. Conclusion and future work

In this paper we introduced a methodology for an automatic detection of improvable diagnosis functions which have to be updated. Since the evaluations proved very promising, we integrated it into the reports for field data which are part of BMW's continuous improvement process for automotive diagnosis.

Our next goal is to include the shown approach into the development process for the off-board diagnosis for all automotive components for an overall evaluation.

Furthermore, we are discussing with the developing departments how to increase the extent of our analysis with the future goal to create a fully automated closed-loop diagnosis feedback process. We are convinced that our approach can be of great benefit to increase the off-board diagnosis' quality thus reducing warranty costs and repair time.

References

[1] Negele, H., "Systems Engineering Challenges and Solutions from an Automotive Perspective. Keynote Presentation," in *INCOSE Symposium 2006*, 2006.

[2] Broy, M., Krüger, I., Pretschner, A., and Salzmann, C., "Engineering automotive software," *Proc. of the IEEE*, vol. 95, no. 2, pp. 356–373, 2007.

[3] Avižienis, A., Laprie, J. C., Randell, B., and Landwehr, C., "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Trans. on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004.

[4] "Bürgerliches Gesetzbuch," German Ministry of Justice.

[5] Schäuffele, J. and Zurawka, T., *Automotive Software Engineering*. SAE International, 2005.

[6] Meinzer, S., Prenninger, J., Eberl, M., and Eren, T., "Durch Predictive Analytics von Diagnosedaten zu fundierten Qualitätsmanagemententscheidungen und höherer Kundenzufriedenheit," in *Text- und Data Mining für die Qualitätsanalyse in der Automobilindustrie, Leipziger Beiträge zur Informatik Vol XXV*, Heyer, G., Luy, J.F., and Jahn, A., Eds., 2010, pp. 17 – 27.

[7] ISO, "ISO 15031-6 Road vehicles – Communication between vehicle and external equipment for emissions-related diagnostics - Part 6: Diagnostic trouble code definitions," 2005.

[8] Buddhakulsomsiri, J. and Zakarian, A., "Sequential pattern mining algorithm for automotive warranty data," *Computers & Industrial Engineering*, vol. 57, no. 1, pp. 137–147, 2009.

[9] Blumenstock, A., Schweiggert, F., Müller, M., and Lanquillon, C., "Rule cubes for causal investigations," *Knowledge and Informations Systems*, vol. 18, no. 1, pp. 109–132, 2008.

[10] Müller, T., Krieger, O., Lange, K., Breuer, A., and Form, T., "Neuronale Netzmengen für die Fehlerdiagnose in komplexen Fahrzeugsystemen," in *Diagnose in mechatronischen Fahrzeugsystemen*, Bäker, B.A. and Unger, A., Ed., 2009.

[11] Sankavaram, C. *et al.*, "Event-driven Data Mining Techniques for Automotive Fault Diagnosis," in *Proc. of the 2010 Internat. Workshop on Principles of Diagnosis (DX 2010)*, 2010.

[12] Tabachnick, B. and Fidell, L., *Using Multivariate Statistics*, 5th ed. Allyn and Bacon, 2006.

[13] Jain, A. K. and Dubes, R. C., *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[14] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J., *Classification and regression trees*. Chapman and Hall, 1984.

[15] Agrawal, R. and Srikant, R., "Fast algorithms for mining association rules," in *Proc. of the 20th Internat. Conference on Very Large Data Bases*, 1994, pp. 487–499.

[16] Nikulin, M.S., "χ^2-test for continuous distributions with scale and shift parameters," in *Theory of Probability and its Applications*, vol. 18, no. 3, 1973, pp. 559–568.

[17] Dorneich, A., "Interactive Analyzer," http://www.i-analyzer.com.

[18] Prestl, W., Sauer, T., Steinle, J., and Tschernoster, O., "The BMW active cruise control (ACC)," *SAE Trans.*, vol. 109, no. 7, pp. 119–125, 2000.