Forecasting the Fault Rate Behavior for Cars

Guido Lindner
DaimlerChrysler AG
Research and Technology FT3/KL
po: DaimlerChrysler AG, T-402
D-70456 Stuttgart, Germany
Email: guido.lindner@daimlerchrysler.com

Rudi Studer
Institute AIFB
University Karlsruhe
D-76128 Karlsruhe, Germany
Email: studer@aifb.uni-karlsruhe.de

Abstract

This paper describes the development of a data mining application delivering a trend prognosis of the fault rate behavior of cars. We discuss how a carefully chosen preprocessing strategy, which uses hierarchical clustering, results in a transformation and restructuring of the given fault behavior data. The transformed data are the basis for the model generation process which uses either C4.5 or neural networks. The trend prognosis system is implemented as a prototype which uses the KDD-Tool CLEMENTINE[1]. This prototype has been evaluated in a field test. When compared with the conventional approach which has been used up to now, our approach provides trend prognoses which are available several months earlier and delivers results which are more accurate.

1 Introduction

Quality control of the produced cars[2] is a major topic in the car industry. One part of a quality control program is the systematic collection of information about the fault behavior of the sold cars. Within DaimlerChrysler AG the Quality Information System (QUIS) is used for this purpose. In QUIS all production information about each by Mercedes produced car is stored. Additionally QUIS includes all information about the warranty cases and garage visits of the sold cars. The QUIS system is used for various tasks, like

- prediction of warranty costs,
- analysis of customer complaints,
- product observations like quality reports,
- detection of complaints which are different from the calculated *standard*[3]

Currently, the database contains information of more than seven million cars and the database has a size of 40 giga bytes.

Today, the QUIS system provides means to observe special complaints and to compute a trend prognosis for the current month based on the relative number of complaints from the last two months. This kind of analysis is illustrated in figure 1. In figure 1 show the x axis shows the production month and the y axis the quote of complaints. Each line represents the complaint rate for a special time period, e.g. three months, six months and nine months after the production month are usual observation points. It is obvious that each warning about a negative trend which can be detected earlier can save a lot of money and can avoid a damage of the reputation of the company. For example, if a quality problem is detected for cars, which are six months old, the production problems already exists for six months. If one could develop a system which can deliver the same information two or three months earlier, one could save a lot of trouble with customers and of course a lot of money.

In this paper we describe our approach to improve the trend prognosis for the fault behavior of sold cars. Special emphasis is put on detecting trends as early as

[1] Trademark from ISL

[2] In this paper we use cars as a synonym for all kinds of automobiles.

[3] All components have an expected rate which is normal and is a result of the normal production process

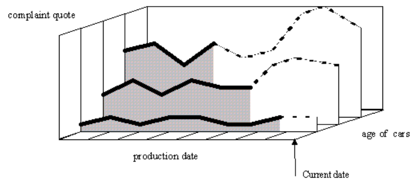

Figure 1: Analysis in QUIS

possible. For this purpose we developed a data mining application which uses machine learning and statistical techniques. The developed prognosis system is currently used in a field test. This test shows that the system delivers more accurate results much earlier than the prognosis system which is in use up-to-now. Thus, DaimlerChrysler can react to evolving quality problems much faster and can thus reduce costs and improve customer satisfaction.

The rest of the paper is structured as follows. Following the main steps of the KDD-process model from CRISP[4] [Chapman et al., 1998](see figure 2) we analyze in section 2 the application and outline our approach to get a *good* trend prognosis earlier than with the *conventional system*. This includes a description of the pre-processing steps which are crucial for the success of our approach. In section 3 we discuss the conducted experiments and results. In the last section we validate the results of our approach and give a conclusion as an interpretation and evaluation step of the process model.

2 Application Analysis

Section 2.1 describes the basic steps to develop an understanding for the application and also outlines former approaches for providing a trend prognosis. The following section 2.2 describes an analysis of the domain data according to the broadly outlined step of data understanding [Chapman et al., 1998]. This section is finished by the definition of the pre-processing (data preparation) step.

[4]The CRISP KDD-process model is compatible to the process model from [Fayyad et al., 1996]

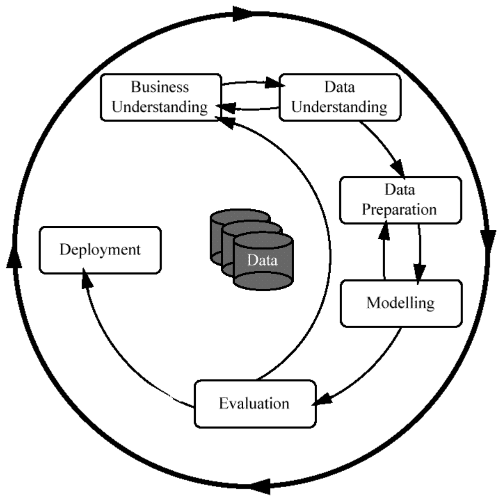

Figure 2: CRISP Process Model

2.1 Former Work

A first study [Legner et al., 1996] aimed at developing a system to predict the warranty costs of cars. That study used several time series strategies for computing a prognosis of the warranty costs. Although this approach failed in the end, one could get an important insight from this study: The curves describing the complaints of cars which are of different age, are rather similar, but located on different rate levels. I.e., the real rates differ, but the patterns of the curves are quite similar. With the help of this observation [Legner et al., 1996] chose a strategy to forecast the warranty costs of the cars per production month. In difference to the analysis method used in the QUIS-System (see figure 1), they did not only use the production time. Additionally, they used the age of the cars (see figure 3) and got much better results with a twofold linear regression to forecast the costs, first vertical to forecast the quote of complaints for a production and then horizontal the productions in the future.

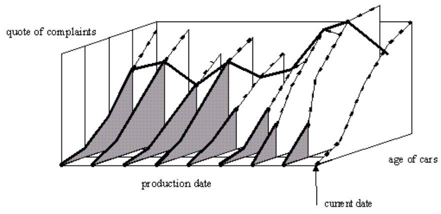

Figure 3: First approach

In this first study the life time and the production time are used separately from each other and so the information about the interdependency between these two types of information is lost. In our approach we use all available data in an integrated way. Figure 4 shows the basic idea. All data in the grey planes are known and are used for the trend prognosis. Thus, the interdependencies between the different kinds of information are exploited. Furthermore, our approach is different to the one described in [Legner et al., 1996] in another important aspect. We want to forecast the fault behavior of cars and not the warranty costs. This task is more difficult than forecasting the sum of warranty costs: when forecasting faulty behavior, one is more interested in special details like the behavior of one component or a group of components.

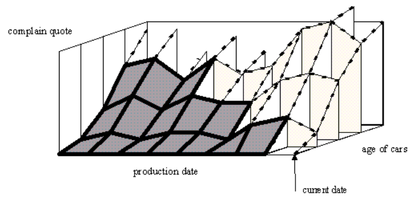

Figure 4: Our Approach

2.2 Domain data

For our application task we have to analyze the available data. Although, there are a lot of data available in the QUIS database, a careful analysis shows that QUIS does not contain the required data for the subsequent model generation step. This situation results in the specific pre-processing step which was chosen for our approach.

All garages have the instruction to encode the complaints with the help of a code book. Today there are more than 5000 components of a car which can be the cause of a complaint. The code book is structured in function groups. These function groups are based on construction codes. This is to say, that the function groups are derived from the construction codes. But the structure is quite different. A construction group describes the main parts of a car like the engine or the doors. A function group can contain elements of different construction groups. These function groups contain pictures which represent a certain functionality of the car, so that the mechanic can easily assign the cause of a complaint to a function group to get the code of the causing component. Each component has a unique code and is exactly part of one construction group, but can be part of several function groups. For example the construction group of Brakes is called 42 and the function group Operation Brake 42.10. The component 42.104 is in the function group Operation Brake, but is also a member of the function group 03.20 Engine Electric.

The selection of the raw data contains several megabytes, because every complaint of a car is one entry in the QUIS database. But we need a representation form based on the components, which are causing the complaint.

However, the first pre-processing step results in a problem: we draw a sample of 54000 cars from one production year and the history of complaints of two years. Out of the 5000 possible causes of complaints only 2600 were used in the sample and only 600 had a rate higher than 0.1% . We chose the limit of 0.1% because this rate requires more than 54 complaints per year worldwide. With a rate less than 0.1% we get less than five complaints per month on the average and we can suppose that these complaints are accidental. These 600 components are members of 50 construction groups and 150 function groups. If we require a rate of complaints of 0.1% in the first quarter of the year, we get complaints in only 40 construction and 90 function groups. Obviously, these few cases are not sufficient for a model generation step.

Therefore, we decided to cluster the complaints in the pre-processing step. Such a clustering step has several advantages with respect to the given QUIS database:

- we get larger training sets for the model generation,
- groups with more complaints have a better statistical reliance,
- we get some homogeneous clusters.

But there are some more benefits with respect to the given application task:

- Clustering into useful groups reduces the complexity and increases the possibility for the experts to interpret the results.
- Clustering can reduce the problem of wrong settings by the garages. If they take the wrong code for the causality component of a complaint, the right component has a local or functional relation to the wrong coded component with a high possibility.

- Changes in the clusters may be a good indicator for a new kind of complaint.

In the end, we had to deal with a rather surprising situation. Given a database with giga bytes of training data, the first analysis and pre-processing step did not offer sufficient training cases for the model generation phase. This is in contrast to the *classic* discussion of how to get rid of too many training cases. Instead of reducing the set of available training cases, we use clustering to build up larger training sets. However, we think that this situation will arise in other application scenarios as well.

In the next section we describe the selected clustering and the data mining algorithms and discuss the achieved results.

3 Data Preparation and Modelling

3.1 Clustering

The clustering step aims at grouping single components, i.e. their codes, into meaningful clusters. Therefore we had to define what we mean by *meaningful clusters*. A meaningful cluster groups parts that are functionally dependent or that are located near to each other in the car ("local dependencies"). If such a group of parts is found conspicuous, this gives valuable information for finding the real cause of the complaint.

Unfortunately, it is not obvious of how to derive clustering attributes for the components that show functional or local dependencies. We identified three potentially useful criteria for clustering the component codes:

- **fault rate behavior over lifetime of vehicles:** The complaints in the first year in the lifetime of all of the cars are analyzed. Parts are found to be similar if they have a similar behavior, e.g. tend to break early or late.
  We recorded the cumulated complaint quotes of the parts in intervals of 7 days. We took these numbers as attributes and calculated a cosine distance measure, which is independent of scaling and compares rather the form of the curve described by the datapoints [Gordon, 1981].

- **synchronous changes of fault rates:** The fault rates are traced over the production period of the cars. If the fault rates of two parts rise or fall simultaneously, there may be a conjoint cause indicating a functional dependence.
  The given year of production is divided into 2 week intervals. For each interval the total number of complaints for the one year old cars is determined. The distance is calculated as the correlation between these data points of two parts. This again is independent of the scaling, but in contrast to the cosine also independent of the mean.

- **background knowledge from code book:** As the other criteria are not to distinctive, it seems reasonable to involve further background knowledge from the code book. This book represents the experts' knowledge mainly about local, but also about functional dependencies. The hierarchy used in this book is composed regardless of homogeneous group sizes or the special aim of common prediction, so it can only be used as additional information.
  We chose a simple match as distance measure, which means, that we have a distance of 0 for components of the same construction and function group and values greater than 0, if the function or even the construction group is different. Alas, this simple and feasible measure ignores for example similarities of adjacent construction groups.

As we did not aim at building clearly separated clusters, we did not intend to find a partition of all of the component codes into distinct groups via partitioning cluster analysis. Instead we built a hierarchy out of the codes with several levels of similarity.
We used hierarchical agglomerative clustering algorithms, which are well known and investigated [Gordon, 1981, Mucha, 1992]. These algorithms take a number of data points, a distance measure between them and a fusion strategy and work by successively agglomerating the nearest data points. For this cluster building process, the distance measure was composed as the weighted sum of the three criteria mentioned above. We decided to take the flexible weighted agglomeration strategy of Lance and Williams for the successive fusion of the single parts and groups [Lance and Williams, 1966]. This offers the possibility to easily assess the quality of the cluster process by slightly varying the parameters.

We used the distance of the component codes with very small weights, so that the hierarchy is slightly directed to the existing component hierarchy, but nevertheless, parts from different function groups are not excluded from being put into one cluster. These runs yield the best results as most of the groups are comprehensible for the experts, but the influence of fault rate behavior and the synchronous changes of the rates is still quite high.

3.2 Forecast of fault behavior

For implementing our application, we use the KDD-Tool CLEMENTINE. We added a special *QUIS functionality* to CLEMENTINE for adapting the QUIS data to the different steps of our approach:

- clustering
- training step
- use and test step

Additionally, we included a cluster function into CLEMENTINE, which is not part of the standard version as part of the CITRUS project [Wirth et al., 1997]. With these additional functions we built a system according to the different process steps which are described above. We then evaluated the approach with different algorithms of CLEMENTINE.

We used the groups found in the clustering phase to test our forecasting approach. We made no difference between single parts and groups of parts, as long as their total number of complaints was in a reasonable range. That is, the number of complaints is big enough to have some statistical relevance, but especially for groups they are not higher than a certain fraction of the total number of complaints (we chose 20% here). About 350 parts and clusters fulfilled these criteria.

Figure 5 show the attributes that describes a single group at a given month of production. The x-axis represents the date of production, E stands for the given month, A to D are the 4 preceding months. For the cars of each period the relative complaint quotes are recorded in intervals of 14 days and plotted on the y-axis. Each of the points in figure 5 is named by using the period of production and the age of the cars as an index, e.g. D_{84} is the quote of complaints on the given component(s) that occurred for the cars produced one month ago (relative to the given month) until they are 84 days old. Notice that the reference month and anything like a name for a component or a group are not included in the data, as we tried to learn a common behavior of the parts given only the numbers representing the behavior in the past. If we assume, that today is one month after the reference month E, we know the first two quotes E_{14} and E_{28} of the last month. Accordingly, we know D_{14} to D_{56}, C_{14} to C_{84} and so on, whereas the numbers in the upper right part of figure 5, as separated by the diagonal line, are still unknown and may not be used as input variables for prediction. For forecasting of the 6, 9 and 12 month trends we lift up this knowledge triangle.

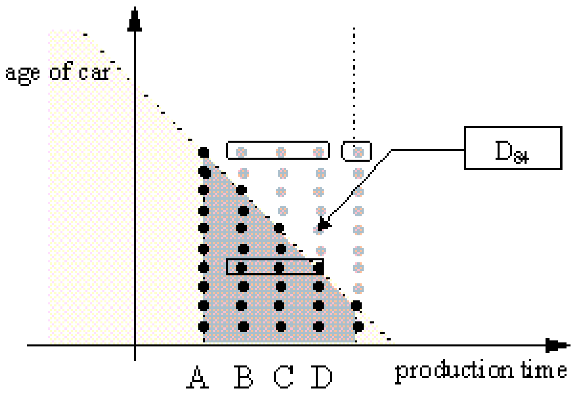

Figure 5: Datapoints for the prognosis

In a first experiment we compared regression and classification. The experiment showed that classification delivers better results and that the classification models are more accepted by the users of the prediction system.

For classification we chose two thresholds to separate the calculated complaint quote trends into three classes "falling", "constant" and "raising", with about $\frac{2}{3}$ of the occurring quotes being classified as "constant" (which rather means "not conspicuous" than really constant). Predicting these symbolic classes becomes a classification problem, which we solved by applying $C4.5$ [Quinlan, 1993] as well as a neural network.

To assess the quality of classification we defined two measures q_1 and q_2 as follows :

$$q_1 := \frac{Number\ of\ correctly\ classified\ conspicuous\ components}{Total\ number\ of\ conspicuous\ components}$$

$$q_2 := \frac{Number\ of\ correctly\ classified\ conspicuous\ components}{Number\ of\ components\ classified\ as\ conspicious}$$

q_1 specified the completeness and q_2 the correctness of the prediction. In the information theory we find similar measurements like recall and precision. Especially q_2 is important for this application because every part of the car that is classified as conspicuous has to be investigated by an expert. False alerts will therefore cause unnecessary efforts and costs and therefore the number should be kept low.

Input	C4.5	neural network
performance at trainingset	$q_1 = 0.635$ $q_2 = 0.723$	$q_1 = 0.588$, $q_2 = 0.808$
performance at testset	$q_1 = 0.584$ $q_2 = 0.665$	$q_1 = 0.578$, $q_2 = 0.811$

Table 1: Prediction of classes

In our experiment the neural network performs better than $C4.5$. One main reason for learning only one model for all parts and all months was to get good generalizing capabilities. Table 1 shows the results of another experiment where we tried to prove this capability. We presented only data from the first 6 month of the production year for learning and used the other half of the year for assessment purposes. As one can see, the quality does not change significantly for the neural network, whereas $C4.5$ has obvious problems with generalizing.

Compared to the prognosis system which is currently in use, the best neural network found 65% instead of 60% of the conspicuous parts (q_1), with only 20% instead of 30% false alerts (q_2); in addition, these results are delivered three months earlier. This is a considerable improvement of the trend prognosis process. We made additional tests with the neural network with different parameters. All results of these test are in the same region (see figure 6).

The prediction for the forecasting of the six, nine and 12 month curves are quite better (+10%), but the prediction of the three month curve was one of the main goals of the project.

In a next step we selected two special parts and predicted the complaint behavior as a pilot application. Based on rather successful results, the product controlling department decided to evaluate our system for all cars in their daily work.

4 Conclusion

In this paper, we introduced the data mining application we developed within DaimlerChrysler AG for predicting the fault rate behavior of cars. According to the process model of CRISP we divided the development process into different phases.

Although we had available a database with several giga bytes of data, the analysis of the data in the pre-processing phase showed that in the end not enough training cases were available in the raw data. We therefore used a clustering algorithm in the pre-processing phase to build up clusters of components which are then used to learn the prognosis model. In that way, the pre-processing step is to some extent a knowledge discovery step as well [Engels et al., 1997]. We think that this approach is not specific for our company, rather we believe that this approach might be useful in a large variety of quality control applications for technical products.

Our implemented system was successfully tested in real-life applications and has now been transferred to the product controlling department. Our system delivers a more accurate trend prognosis much earlier when compared with the system which is in use up-to-now. This shows that a carefully designed data mining application can significantly improve the handling of crucial business tasks.

Acknowledgments

We thank Mr. Litwin and Mr. Vennebörger from DaimlerChrysler AG for their support in developing the data mining application. We are also indebted to the ML–group of the DaimlerChrysler Research and Technology FT3/KL for fruitful discussions. Special thanks also to Udo Spuhl who implemented the pre-processing functionality.

References

[Chapman et al., 1998] Chapman, P., Clinton, J., Hejlesen, J. H., Kerber, R., Khabaza, T., Reinartz, T., and Wirth, R. (1998). The Current CRISP-DM Process Model for Data Mining. http://www.ncr.dk/CRISP/index.html.

[Engels et al., 1997] Engels, E., Lindner, G., and Studer, R. (1997). A Guided Tour through the Data Mining Jungle. In Heckerman, D., Mannila, H., Pregibon, D., and Uthurusamy, R., editors, *Proc. of the 3rd International Conference on Knowledge*

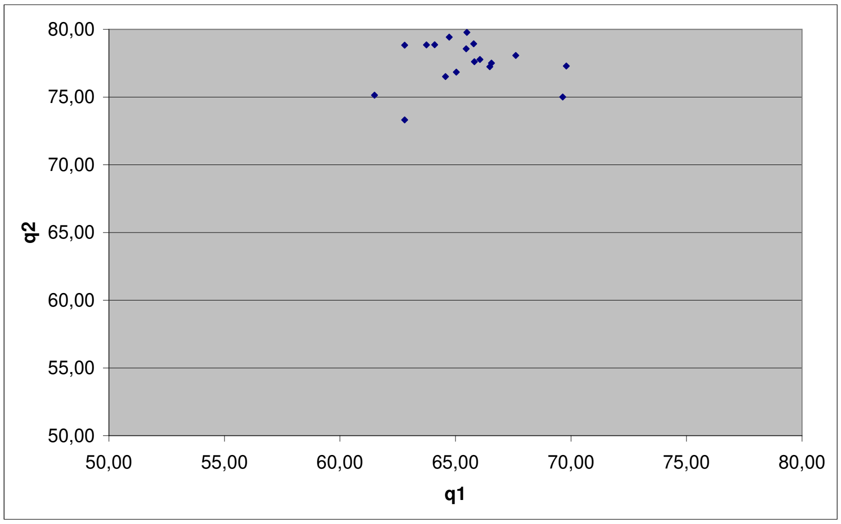

Figure 6: Results to predict the trend behavior for 3 months curve

Discovery and Data Mining, pages 163 – 166. AAAI Press.

[Fayyad et al., 1996] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). Knowledge Discovery and Data Mining: Towards a Unifying Framework. In Simoudis, E., Han, J., and U.Fayyad, editors, *Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 82 – 88, Portland, Oregon.

[Gordon, 1981] Gordon, A. (1981). *Classification*. Chapmann and Hall, London.

[Lance and Williams, 1966] Lance, G. and Williams, W. (1966). A General Theory of Classificatory Sorting Strategies. *Computer Journal*, 9:373–380.

[Legner et al., 1996] Legner, C., Nakhaeizadeh, R., and Ohl, S. (1996). Mining in Warranty Cost Data. In *Proc. of the Thirteen European Meeting on Cybernetics and System Research*.

[Mucha, 1992] Mucha, H. (1992). *Clusteranayse mit Mikrocomputern*. Akademie Verlag, Berlin.

[Quinlan, 1993] Quinlan, J. (1993). *C4.5: Programs for Maschine Learning*. Morgan Kaufmann, San Mateo, CA.

[Wirth et al., 1997] Wirth, R., Shearer, C., Grimmer, U., Reinartz, T., Schloesser, J., Breitner, C., Engels, R., and Lindner, G. (1997). Towards Process–Oriented Tool Support for KDD. In *Proceedings of 1st Intern. Conference on Principles of KDD*. Springer Verlag. LLNCS Serie.