

Application of Bayesian Partition Models in Warranty Data Analysis

Markus Mueller*

Christoph Schlieder †

Axel Blumenstock ‡

Abstract

Automotive companies are forced to continuously extend and improve their product line-up. However, increasing diversity, higher design complexity, and shorter development cycles can produce new and unforeseen quality issues. Warranty data analysis helps quality engineers in their task of identifying the root cause of manufacturing or design related problems and in planning and implementing remedial actions. In this paper we show how Bayesian partition models can be used to support root cause investigations by applying Bayesian model comparison. We review product partition models, exemplify how partitions can be ranked, and illustrate their expressive power compared to Bayesian networks. Based on this, we outline a data analysis approach that considers dependencies, in particular taxonomic and partonomic relationships, among influencing variables and identifies the most likely semantically meaningful partitions that are close to the concept that actually caused a quality issue. The approach can be integrated seamlessly with interactive decision trees which have been successfully applied in our domain. An evaluation on test data and real-world case studies illustrate how the approach can be used by engineers to investigate cause-effect relationships and show that its application is not limited to the automotive domain.

1 Introduction

Modern vehicles are highly complex systems, and although automotive companies continuously improve their design and manufacturing related processes, unforeseen quality issues can arise. To reduce warranty costs and to improve customer satisfaction, quality engineers have to detect and resolve quality issues as fast as possible. A system for warranty data analysis has to support engineers in their complex task of investigating root causes of quality issues. When the true cause is uncovered, the system should further help in planning and implementing effective and cost-efficient actions to fix this problem and to prevent this or similar problems in future. Such an approach is not only useful in the automotive domain. Whenever observations should be turned into actions, knowledge about cause-effect relationships is necessary.

1.1 Problem Description The main problem is to (semi-)automatically identify a small subset of many

variables that might have influenced a quality issue. These variables provide information about vehicle design, production, or the failure context. The quality issue is encoded as binary class variable that separates so-called non-conforming vehicles from all other vehicles produced.

A general data mining approach to this problem comprises the following generic sub-tasks: Pattern search, assessment, and visualization. The approach has to identify variables and combinations of variables that are statistically associated with a given (binary) target variable. The patterns found must be ranked and statistically validated. This process should be highly interactive to allow the user to incorporate his background knowledge to derive *actionable* knowledge [2]. As users are not data analysis experts, process and results must be easily understandable.

The following specifics have to be considered in our application scenario: First, the distribution of the class variable is notably skewed as the number of non-conforming vehicles is generally small. Second, data quality is poor. In particular, the target variable might be very inaccurate in describing the set of non-conforming vehicles. Moreover, dependencies among describing variables can cause misleading influences showing up. Simpson's paradox can also occur and truly multivariate patterns might be missed. What is worse, the true cause is often hidden. Hence our goal is to reach the true cause as close as possible by suppressing findings that are likely to be non-causal. Semantic hierarchies like taxonomies or partonomies provide useful information to support this goal.

In this paper we present an approach that addresses these issues. It is based on the application of Bayesian partition models [11, 1, 6].

1.2 Outline Our paper is organized as follows: In the next section we first revise the idea of Bayesian model comparison [19] and the work of Hartigan and Barry [11, 1] on product partition models. We define interestingness measures to rank partitions and demonstrate the expressive power of partition models by comparing them to Bayesian networks [12, 7]. In section 3 we describe an interactive data analysis approach that applies Bayesian partition models to support root cause

*Laboratory for Semantic Information Technology, University of Bamberg, Bamberg, Germany

†Chair for Computing in the Cultural Sciences, University of Bamberg, Bamberg, Germany

‡Quality Analysis, Daimler AG, Stuttgart, Germany

investigations. We evaluate our approach on test data and illustrate its overall use in a real-world case study in section 4. Related work is discussed in section 5.

2 Model Comparison using Bayesian Partition Models

Model comparison denotes the task of finding the model fitting the data best among various models with different parameter sets [19]. In this section we revise this concept and its application to Bayesian partition models. We present *product partition models (PPMs)* developed by Hartigan and Barry [11, 1] as one form of partition models. Besides, we illustrate how partition models are related to Bayesian networks.

2.1 Bayesian Model Comparison Model comparison is difficult because complex models, i.e. models with a high-dimensional parameter vector θ , can fit data better than simpler models. On the other hand, overly complex and over-parametrized models do not show adequate generalization behavior (overfitting). Bayesian model comparison aims at optimizing this trade-off [19].

Applying Bayes' rule, the posterior probability of a model \mathcal{H}_i can be derived from its *marginal likelihood* or *evidence* $p(\mathcal{D}|\mathcal{H}_i) = \int p(\mathcal{D}|\theta_i, \mathcal{H}_i) f(\theta_i|\mathcal{H}_i) d\theta_i$, the *prior probability* $p(\mathcal{H}_i)$ and $p(\mathcal{D}) = \sum_j p(\mathcal{D}|\mathcal{H}_j) p(\mathcal{H}_j)$:

$$p(\mathcal{H}_i|\mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{H}_i) p(\mathcal{H}_i)}{\sum_j p(\mathcal{D}|\mathcal{H}_j) p(\mathcal{H}_j)}.$$

Two models can be compared by evaluating the *posterior odds ratio*, but MacKay points out that Bayesian model comparison naturally embodies Occam's Razor even without penalizing complex models by assigning low prior probabilities $p(\mathcal{H}_i)$ [19]. Assuming equal prior probabilities for different models \mathcal{H}_i yields the *Bayes factor*:

$$\text{Bf}_{1,2} = \frac{p(\mathcal{D}|\mathcal{H}_1)}{p(\mathcal{D}|\mathcal{H}_2)} = \frac{\int p(\mathcal{D}|\theta_1, \mathcal{H}_1) f(\theta_1|\mathcal{H}_1) d\theta_1}{\int p(\mathcal{D}|\theta_2, \mathcal{H}_2) f(\theta_2|\mathcal{H}_2) d\theta_2}.$$

Bayes factors play a key role in Bayesian model selection: "The Bayes factor is a summary of the evidence provided by the data in favor of one scientific theory, represented by a statistical model, as opposed to another" [16]. There exist practical guidelines for the interpretation of Bayes factors on the log-scale (see tables in [14, 16]). One of the main advantages of Bayes factors with respect to hypothesis testing in frequentist statistics is that a Bayes factor $\text{Bf}_{1,0}$ that compares an hypothesis \mathcal{H}_1 to the null hypothesis \mathcal{H}_0 does not only provide evidence *against* \mathcal{H}_0 , but can also provide evidence *in favor of* \mathcal{H}_0 . Kass and Raftery also point out that frequentist tests tend to "systematically reject null

hypotheses" when the sample size gets very large [16]. Bayes factors do not share this property, which is very important in our domain where the sample size can easily exceed hundreds of thousands of vehicles.

Bayes factors are quite sensitive to the choice of prior parameter densities $f(\theta_i|\mathcal{H}_i)$. The *Schwarz criterion* S is a rough approximation to the logarithm of the Bayes factor without need to specify these densities [16]:

$$(2.1) \quad S = \log \left(p(\mathcal{D}|\hat{\theta}_1, \mathcal{H}_1) \right) - \log \left(p(\mathcal{D}|\hat{\theta}_2, \mathcal{H}_2) \right) - \frac{1}{2} (d_1 - d_2) \log(n),$$

where $\hat{\theta}_i$ is the *maximum likelihood estimate* for θ_i under \mathcal{H}_i , d_i is the dimension of θ_i , and n is the sample size. As $n \rightarrow \infty$, the Schwarz criterion satisfies

$$\frac{S - \log(\text{Bf}_{1,2})}{\log(\text{Bf}_{1,2})} \rightarrow 0.$$

The well-known *Bayesian information criterion (BIC)* is minus twice the Schwarz criterion.

2.2 Product Partition Models Product partition models [11, 1] are used to partition a set of n observations x_i into subsets S_k , where observations belonging to the same subset are assumed to be exchangeable and observations from different subsets are assumed to be independent. The main idea is that if the probability distribution of random partitions is in a certain product form prior to making the observations, it is also in product form given the observations [1].

The prior probability of a partition g consisting of $d(g)$ blocks S_k is given by

$$p(g) = K c(S_1) \cdots c(S_k) \cdots c(S_{d(g)}),$$

where each $c(S_k)$ is a nonnegative prior *cohesion* that is assigned to the vector of random variables $\mathbf{X}_k = [X_{a+1}, \dots, X_b]$, where $0 \leq a < b \leq n$. The constant K is chosen so that the sum over all partitions is unity.

A probability function $p_{\mathbf{X}_k}(\mathbf{x}_k|\mu_k)$ is assigned to each block S_k with all variables $X_i \in \mathbf{X}_k$ being dependent from the same parameter μ_k . With $f(\mu_k)$ being the prior density of the block parameter μ_k and assuming that the X_i of a block are independent given a parameter μ_k , the probability function of the random variables \mathbf{X}_k is given by

$$(2.2) \quad p_{\mathbf{X}_k}(\mathbf{x}_k|g) = \int_0^1 \left[\prod_{i=a+1}^b p_{X_i}(x_i|\mu_k) \right] f(\mu_k) d\mu_k.$$

Based on the additional assumption that given a specific partition and given the parameters, the observations of

different blocks are independent, we get the marginal likelihood for a partition g given the data vector \mathbf{x}

$$(2.3) \quad p(\mathbf{x}|g) = \prod_{k=1}^{d(g)} p_{\mathbf{X}_k}(\mathbf{x}_k|g),$$

and its posterior probability

$$p(g|\mathbf{x}) \propto \prod_{k=1}^{d(g)} p_{\mathbf{X}_k}(\mathbf{x}_k|g) \left(K \prod_{k=1}^{d(g)} c(S_k) \right).$$

In our application we assume that the random variables X_i are Bernoulli distributed, i.e.

$$p_{X_i}(x_i|\theta_i) = \begin{cases} \theta_i, & \text{if } x_i = 1 \\ 1 - \theta_i, & \text{if } x_i = 0 \end{cases}$$

This follows from the simplifying assumption that the i -th vehicle in the set of all vehicles produced has a chance θ_i of belonging to the group of non-conforming vehicles. All $X_i \in \mathbf{X}_k$ share the same parameters μ_k which are assumed to follow a Beta distribution with hyperparameters m_{k_1} and m_{k_2} :

$$\mu_k \sim \text{Beta}(m_{k_1}, m_{k_2})$$

Inserting this into equations 2.2 and 2.3 leads to

$$(2.4) \quad p(\mathbf{x}|g) = \prod_{k=1}^{d(g)} \frac{\text{beta}(m_{k_1} + x_k, m_{k_2} + n_k - x_k)}{\text{beta}(m_{k_1}, m_{k_2})},$$

where $x_k = \sum_{X_i \in \mathbf{X}_k} x_i$, and n_k is the number of X_i in \mathbf{X}_k (see [1]). $\text{beta}(\alpha, \beta) = (\Gamma(\alpha)\Gamma(\beta))/\Gamma(\alpha + \beta)$ denotes the Beta function and we get:

$$p(\mathbf{x}|g) = \prod_{k=1}^{d(g)} \frac{\Gamma(m_{k_1} + m_{k_2})}{\Gamma(m_{k_1} + m_{k_2} + n_k)} \prod_{i=1}^2 \frac{\Gamma(m_{k_i} + N_i)}{\Gamma(m_{k_i})},$$

where $N_1 = x_k$ and $N_2 = n_k - x_k$. The marginal likelihood of a partition is the same as the inner product of the BD metric (see [12]). Imposing constraints on the hyperparameters m_{k_i} yields the BDe metric. For example, setting $m_{k_1} = m_{k_2} = \frac{m}{2d(g)}$, where m is called the *equivalent sample size*, leads to the BDeu metric (see [12]). Setting the hyperparameters of the Beta distribution to $m_{k_1} = m_{k_2} = 1$, i.e. assuming a non-informed prior distribution, yields

$$(2.5) \quad p(\mathbf{x}|g) = \prod_{k=1}^{d(g)} \frac{\Gamma(x_k + 1)\Gamma(n_k - x_k + 1)}{\Gamma(n_k + 2)}$$

which is the inner product of the K2 metric (see [7]). These equivalences will be used to distinguish various PPMs, e.g. PPM(K2).

Several suggestions have been made about how to set the proper prior cohesions [1, 6, 8]. We follow the simple approach described in [15] and set $c(S_k) = \alpha$, $\alpha \geq 0$, which results in $p(g) \propto \alpha^{d(g)}$. Setting $\alpha < 1$ penalizes partitions with many subsets S_k . The parameter α can be adjusted iteratively in an interactive analysis.

While PPMs assume that the parameters θ_i are identical within each group S_k , Consonni and Veronese [6] propose a hierarchical model that assumes similar, but not necessarily identical parameters. These *hierarchical Bayesian partition models* (HBPMs) can be applied in our data analysis approach to easily account for additional variability.

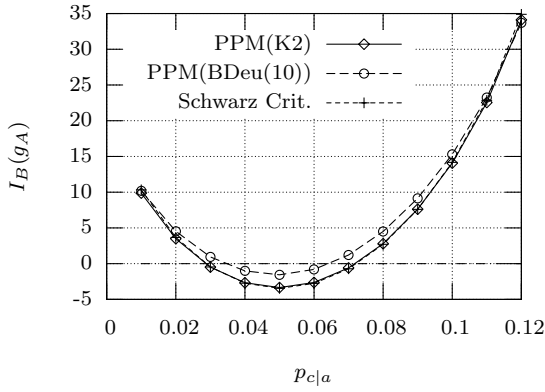
2.3 Assessment of Partitions The interestingness of a partition with respect to another partition can be either assessed by the Bayes factor or by the posterior odds. Following this idea, we define two interestingness measures for a partition g :

$$\begin{aligned} I_B(g) &= \log(\text{Bf}(g; g_-)) \\ &= \log(p(\mathbf{x}|g)) - \log(p(\mathbf{x}|g_-)), \quad \text{and} \\ I_O(g) &= \log(\text{Posterior odds}(g; g_-)) \\ &= \log(p(\mathbf{x}|g)p(g)) - \log(p(\mathbf{x}|g_-)p(g_-)), \end{aligned}$$

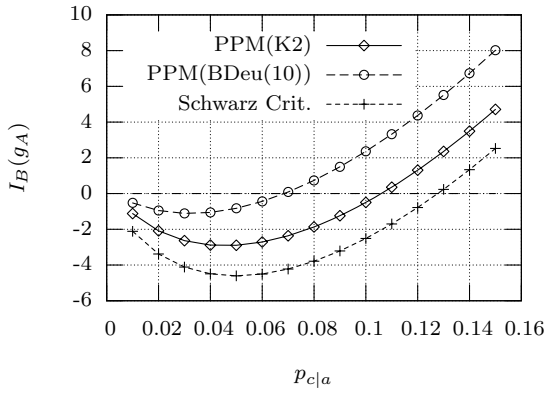
where g_- denotes the prior partition, i.e. the partition with only one parameter. These measures can be used to provide a ranking of partitions and can be applied to compare partitions with each other. In the latter case the prior partition simply cancels out. The partition model and its parameters are given in brackets, e.g. $I_O(g)$ [PPM(K2) ($\alpha = 0.001$)].

The following experiment illustrates how the ranking of partitions changes if we vary the model: The total number of vehicles is 1000 and the prior target share p_c , i.e. the fraction of non-conforming vehicles, is set to 0.05. A skewed class distribution is common in our domain. The binary influence variable A is assumed to be distributed fairly even ($p_a = 0.4, p_{\bar{a}} = 0.6$) or skewed ($p_a = 0.1, p_{\bar{a}} = 0.9$). We gradually increase the target share in a from 0.01 to 0.15. If $p_{c|a} = p_c$, A and C are independent. This means that the prior partition $g_- = \{a, \bar{a}\}$ should be preferred over partition $g_A = \{a\}\{\bar{a}\}$. In this case, one parameter $\mu = p_c$ is sufficient. The more $p_{c|a}$ and $p_{c|\bar{a}}$ deviate from p_c , the more likely partition g_A gets with two parameters $\mu_1 = p_{c|a}$ and $\mu_2 = p_{c|\bar{a}}$.

We use the simple K2 metric (equation 2.5) and the BDeu metric with an equivalent sample size of 10 to assess the likelihood of a partition. Furthermore, we approximate the Bayes factor using the Schwarz criterion (equation 2.1). Figure 1(a) shows $I_B(g_A)$ for



(a) Even distribution: $p_a = 0.4$



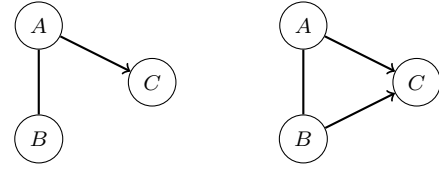
(b) Skewed distribution: $p_a = 0.1$

Figure 1: $I_B(g_A)$ for partition $g_A = \{a\}\{\bar{a}\}$ when the distribution of g_A is almost even or skewed. The target distribution is also notably skewed ($p_c = 0.05$).

these PPMs when $p_a = 0.4$. All models favor g_- when $p_{c|a} = p_c$, but differ in degree of evidence and in regard to the “break even points”, i.e. the $p_{c|a}$ for which g_A becomes more likely.

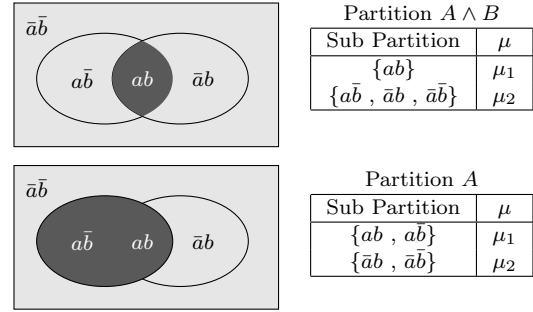
In particular, we can observe an effect recently analyzed by Steck [25]: For large equivalent sample sizes, the BDeu metric suggests a dependence between two variables just because their distributions are notably skewed. Note that the PPM(BDeu) curve lies above the PPM(K2) curve and thus indicates break even points closer to p_c . This behavior would not occur if p_c was set to 0.5. However, this effect becomes even more apparent when the partition gets skewed, too (Figure 1(b)). It is recommended that large equivalent sample size values should be avoided. On the other hand, the Schwarz criterion proves to be a robust approximation, even when distributions get skewed.

Including priors leads to a downshift of the curves. In practice, the parameter α can be set interactively.



(a) Bayesian Networks and CPT of node C

A	B	C	θ	μ_{\wedge}	μ_A
a	b	c, \bar{c}	θ_1	μ_1	μ_1
a	\bar{b}	c, \bar{c}	θ_2		
\bar{a}	b	c, \bar{c}	θ_3	μ_2	μ_2
\bar{a}	\bar{b}	c, \bar{c}	θ_4		



(b) Partition Models

Figure 2: Partition models represent the local dependence structure of Bayesian networks. They allow reducing the number of parameters needed to explain the conditional probability distribution $p(C|Pa(C))$.

Decreasing the start value $\alpha = 1$ iteratively hides noise and helps the engineer in focusing on the strongest correlations.

In general, the experiments show that partitions with large subsets showing strongly deviating target shares are ranked higher than partitions with small subsets or slightly deviating target shares. This is an essential property for scoring functions in subgroup discovery.

2.4 Partition Models versus Bayesian Networks

A Bayesian network represents a joint probability distribution over a vector of n discrete variables $\mathbf{X} = (X_1, \dots, X_n)$ and consists of a *Bayesian-network structure* B_s and a *Bayesian-network probability set* B_p [12]. B_s is a directed acyclic graph (DAG) and encodes a set of conditional independence assertions. Each X_i corresponds to a node in B_s . B_p is a set of local conditional distributions $p_{X_i}(x_i|Pa(X_i))$ for each variable X_i , where the *parent configuration* $Pa(X_i)$ consists of all possible joint states of the *direct* predecessor variables of node X_i in B_s . The *Markov condition* [24, 20] states that given its parent set $Pa(X_i)$, each variable

X_i is conditionally independent of all its other predecessors $\{X_1, \dots, X_{i-1}\} \setminus \text{Pa}(X_i)$. The *d-separation* criterion permits to read off the DAG all (conditional) independencies [20]. These assertions admit the recursive product decomposition

$$p_{X_1, \dots, X_n}(x_1, \dots, x_n) = \prod_i p_{X_i}(x_i | \text{Pa}(X_i)).$$

Bayesian and constraint-based approaches have been developed to learn the structure of Bayesian networks [7, 12, 5, 24, 20]. As already mentioned, assessing the likelihood of a partition is similar to assessing the likelihood of the *local* dependence structure of a Bayesian network. However, partition models can relax the assumption of *local parameter independence* which implies that the parameters that belong to a node in a Bayesian network are independent [12, 10].

Consider Figure 2(a) with three binary variables A , B , and C : The “pseudo” Bayesian network on the left states that A implies C and that B could be correlated with A . No matter whether $B \rightarrow A$, $A \rightarrow B$, or $A \perp\!\!\!\perp B$, in any case $B \perp\!\!\!\perp C | A$, and B is no direct influence of C . The network on the right states that both, A and B are needed to explain C .

Partition models can impose dependencies upon the parameters in the conditional probability table CPT of the target variable C . In PPMs this is realized by considering parameters θ_i exchangeable, in HBPMs the θ_i are assumed to be similar to a common μ_k . It follows that partition models can explicitly model the interaction effect of two variables when influencing a target variable (Figure 2(b)): By combining $\theta_2 - \theta_4$, a logical \wedge can be realized and even the fact that B is not needed to explain the conditional distribution in C , i.e. $p(C|A, B) = p(C|A)$, can be expressed.

Assessing the interaction effect of variables is a very important task in our domain: Engineers want to get detailed information about whether an influence has actually caused a failure or whether it was “pushed” by another influence, whether either of two influences can cause a failure independently, or whether both influences are necessary to cause an issue.

3 Data Analysis using Bayesian Partition Models

In this section we want to present an interactive data analysis approach that is based on the application of Bayesian partition models. The approach is integrated into a productive tool that supports engineers in their task of root cause investigations.

3.1 Overview Bayesian partition models are commonly applied to prediction [1, 8, 15], clustering [23], or

meta-analysis [6, 15]. All these tasks involve some sort of model averaging, e.g. to make inferences about a parameter vector θ . As the number of possible partitions increases faster than exponential with the number of elements, *Markov Chain Monte Carlo* (MCMC) methods [8] or genetic algorithms [4] are commonly applied to enumerate the partitions with the highest posterior probability.

Note that our application context is different: We look for the most likely *semantically meaningful* partitions that are close to the concept that actually caused a quality issue. This is a very important restriction to derive actionable knowledge. Hence, it is not necessary to enumerate all possible partitions. Our goal is to provide a ranking of a *limited* set of partitions that are most likely to hold given the data and that are close to the concept that actually caused a quality issue. We restrict the set of all partitions by a few reasonable assertions:

1. Only the best N partitions are extracted from semantic hierarchies.
2. If a categorical attribute is not part of a taxonomy or partonomy, only the best binary partition is created for this attribute.
3. Only interactions of binary partitions are considered. If a partition contains more than two subsets, the best binary partition is derived from it first.
4. We apply an incremental approach, i.e. interaction effects of partitions are only analyzed for the best binary partitions that result from the previous steps.
5. We only consider interaction effects of a maximum of two partitions.

The approach is integrated with interactive decision trees [22, 2] and can be outlined as follows: The root node refers to the whole dataset and contains the number of non-conforming and conforming instances. Now, the analyst could apply a manual split by selecting a range of the domain of a numeric attribute or by grouping values of a nominal attribute. If he does not have any prior knowledge about the issue, the system creates a ranked list of the best partitions and their interactions—the *partition matrix*. The partition matrix highlights non-causal or similar binary partitions and recommends the next split in the decision tree by looking ahead not only one, but two levels. This can be repeated iteratively, especially when an issue is related to several independent influences. Because of these properties, we refer to the overall approach as *interactive look-ahead decision trees*. Details of algorithm 3.1 will be explained in the next sections.

ALGORITHM 3.1. Interactive Look-ahead Trees

Input: Dataset $(\mathbf{H}, \mathbf{N}, \mathbf{C}, Z)$; Taxonomy T
 \mathbf{H} : hierarchical attributes, \mathbf{N} : numeric attributes
 \mathbf{C} : categorical attributes, Z : class attribute

Output: Decision Tree

```

I[0]  $\leftarrow$  Dataset // instances of root node
while (not userCancel()){
  tempP  $\leftarrow$   $\emptyset$ ; finalP  $\leftarrow$   $\emptyset$ 
   $I \leftarrow$  userSelectSplitNode(I)
   $\forall H \in \mathbf{H}$ : tempP.add(findTopNPartInTax( $I, H, T$ ))
   $\forall N \in \mathbf{N}$ : tempP.add(findBestNumericPart( $I, N$ ))
   $\forall P \in \mathbf{tempP}$ : finalP.add(findBestBinaryPart( $I, P$ ))
   $\forall C \in \mathbf{C}$ : finalP.add(findBestBinaryPart( $I, C$ ))
  matrix  $\leftarrow$  calculatePartMatrix( $I, \mathbf{finalP}$ )
  visualize(matrix) // present matrix and split preview
  selectedPart  $\leftarrow$  userSelectAndPostProcess()
  I  $\leftarrow$  I.split(selectedPart) // grow tree one level
}

```

We want to point out that limiting the number of partition interactions reduces computational complexity and increases understandability, which is both essential for an interactive approach. However, this restriction does not apply to taxonomies. Moreover, if necessary, an arbitrary search depth can be reached as we combine partition models and decision trees. In our experience detecting non-causality is more important than detecting truly higher dimensional attribute interactions.

3.2 Deriving Raw Partitions Partitions on categorical variables can be created very effectively when semantic hierarchies like taxonomies (*is-a*) or paronomies (*is-part-of*) exist. In our application, these are hierarchically structured dimensions (e.g. time, location or vehicle dimension) of a multi-dimensional data warehouse mapped to a relational database by the snowflake model. Applying semantic relationships to support the grouping of domain values is especially valuable as the resulting partitions can be easily interpreted by a user. Consider the model taxonomy in Figure 3. If a problem is only related to vehicles of model C230(Sedan) and C350(Sedan) (indicated by the dark color), all other non-Sedan C-Class models can be grouped together by their body style and all other models can be grouped by their class.

The number of possible partitions is reduced tremendously when only partitions are allowed that exist in taxonomic and paronomic relationships. Nevertheless, this number can still become huge, especially when the tree structure is deep. Therefore, we apply an efficient algorithm that calculates the *top N partitions* among all allowed partitions, exploiting the fact that $p(\mathbf{x}|g)$ is a product of probabilities that can be decomposed recursively (see equation 2.3).

The basic principle of the algorithm can be outlined as follows: Each group S_k knows its subgroups and asks them for the best N (partial) partitions. S_k

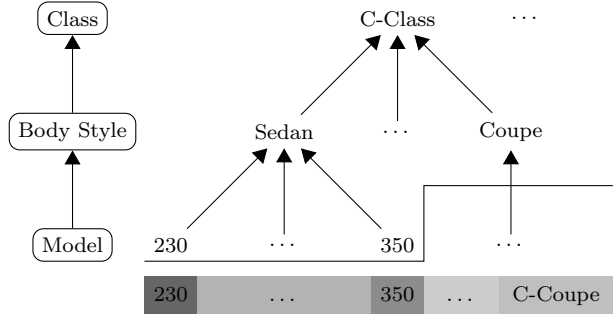


Figure 3: Model taxonomy with a corresponding partition.

merges these partitions, adds a new (partial) partition consisting of itself, sorts the partitions according to their interestingness and finally returns the best N partitions to its predecessor. On the lowest level, each block B_i (e.g. Model=C230(Sedan)) just returns itself as the trivial (partial) partition $g = \{S_k\} = \{B_i\}$. Applying the algebraic rule that $a \cdot b > a' \cdot b'$, if $(a > a' > 0) \wedge (b > b' > 0)$, it can be easily proven that the algorithm finds the best N partitions. When introducing prior cohesions $c(S_k)$, some care is needed: Although the simple cohesions $c(S_k) = \alpha$ fulfill the monotonicity constraint, other prior probabilities $p(g)$ might not.

In case there exists no structural information about the domain of a categorical attribute, we directly search for the best binary partition on its domain. This implies that we have to enumerate all possible binary partitions. For attributes with a large domain we apply an efficient preprocessing algorithm that merges cells with zero counts and very small deviations. Most of the time the number of cells is reduced tremendously by doing so and an exact computation of the best partition is feasible. Up to now there was no need to apply MCMC methods or genetic algorithms as most attributes with many values are part of a hierarchy.

In [18] the authors propose an exact computational procedure to calculate the posterior probability of a partition. The key idea is to exploit the monotonicity of the scoring function by applying dynamic programming. Whenever an order is defined on the attribute domain as it is the case for numeric or ordinal attributes, this approach is very efficient. Apart from that it is often possible to restrict the search space as most interesting patterns for numeric variables refer to binary or ternary splits. Consider, for example, a build date range after a supplier change or a cold temperature range. Preprocessing to reduce the number of possible cut points can be done in linear time.

Proposed partitions may always be edited by users based on their background knowledge so as to achieve

more meaningful groupings. For example, an engineer might split up the build date range by introducing a *clean point*, i.e. a date when the assembly process is changed due to previous quality issues. Nominal groupings may be changed if the user identifies further semantic relationships.

3.3 Partition Matrix A *partition matrix* is used to visualize the interaction effects and the similarity of binary partitions. The reason why we create the best binary partition for each partition first, is simply that one of our main goals is to present results in a way that is intuitive to understand. An “influence” is a binary concept: A specific combination of variable values describes a set of vehicles that is likely to be affected by a quality issue while the contrast set is not. Common interactions of two influences A and B are depicted and explained in Figure 4.

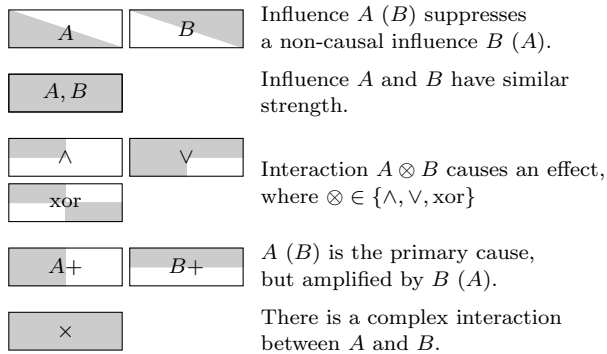


Figure 4: Interactions of binary partitions as they occur frequently in our domain. They are represented by specific icons in the partition matrix and their interestingness defines the fill color of such a symbol on a scale from green to red.

When constructing the partition matrix, we proceed as follows. First, the best binary partitions are created from the raw categorical and numeric partitions. Next, the binary partitions are sorted by their interestingness I . Within each partition, the subset with the highest lift is put up-front. Now, we assess the various partition interactions by calculating I . If no score is positive, the prior partition is the best partition and we assume that there is no dependence. We assume that A suppresses B , if A is the partition that is ranked highest and if $I_B(A) - I_B(B) > \gamma$. On the other hand, A is considered non-causal given B , if B is the best partition and if $I_B(B) - I_B(A) > \gamma$. If the thresholds are not reached in either of these two cases, $[A, B]$ is considered the best interaction. In any other case, we simply output the interaction of partitions that is ranked

highest. The parameter γ is a user-defined threshold that can be set based on Jeffreys’ recommendations about the interpretation of Bayes factors [14, 16]. In our experiments we set $\gamma = \log 100$ which would be considered “strong evidence” of one hypothesis against another one.

Apart from this, the partition matrix also visualizes the similarity of partitions, i.e. whether partitions describe similar subsets of instances. This is very important in our domain with hundreds of variables. A preference of customers for specific packages or technical requirements cause that many sales codes form option clusters that describe similar vehicles. The same applies for weather related variables. Similar influences can often guide users to a hidden influence.

As suggested in [26] for clustering, we apply normalized mutual information (NMI) to measure similarity between two partitions A and B :

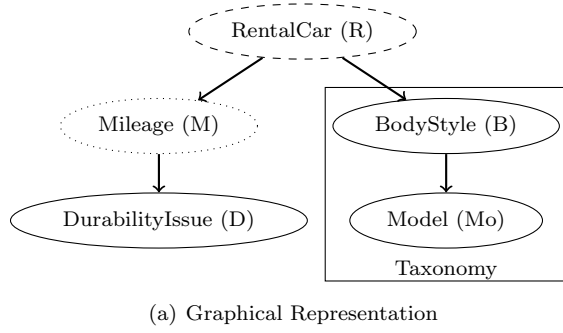
$$\text{NMI} = \frac{I(A, B)}{\sqrt{H(A)H(B)}},$$

where $I(A, B)$ denotes the mutual information between A and B and $H(A)$ denotes the entropy for A . If two partitions are similar, $\text{NMI} \rightarrow 1$. On the other hand, if A and B are independent, $\text{NMI} = 0$. The NMI values are given as percentage and are visualized on a color range from light blue to dark blue in the partition matrix.

3.4 Simulation Study In two simple examples we want to demonstrate how an approach based on the application of Bayesian partition models can properly assess the interaction effects of influences as they occur frequently in our domain. First, consider the scenario depicted as Bayesian network in Figure 5(a).

In the example a durability issue is only related to the variable mileage, which would be a numeric attribute in our application. It is well-known that rental cars or cabs reach high mileage in a shorter time period than other vehicles. This is represented by the influence RentalCar that could also be hidden in practice. On the other hand, large orders of rental car companies often cause deviations in the distributions of models and sales codes. If a huge rental car company leases many vehicles with the same equipment, these sales codes might show up as potential influences although the problem is only related to high mileage. We simulate data containing 10000 instances using Tetrad [21] and run our application on the data.

Figure 6(a) shows a ranking of partitions and the partition matrix for $I_B(g)$ [PPM(K2)]. Mileage is the strongest influence, but a log Bayes factor of 14.6 would consider RentalCar an important finding, too. However,



R	$p(R)$
yes	0.3
no	0.7

R	M	$p(M R)$
yes	high	0.7
yes	low	0.3
no	high	0.2
no	low	0.8

R	B	$p(B R)$
yes	S	0.3
yes	C	0.6
yes	W	0.1
no	S	0.5
no	C	0.2
no	W	0.3

M	D	$p(D M)$
high	yes	0.05
high	no	0.95
low	yes	0.01
low	no	0.99

B	Mo	$p(Mo B)$
S	S230	0.3
S	S280	0.4
S	S350	0.3
C	C230	0.1
C	C280	0.2
C	C350	0.7
W	W230	0.2
W	W280	0.5
W	W350	0.3
...		0

(b) Conditional Probability Tables

Figure 5: Data analysis scenario 1: Mileage is the main influence that causes a durability issue. Other variables might be correlated because of the hidden cause RentalCar.

the partition matrix states that RentalCar only shows up because of Mileage. Similarity between Mileage and RentalCar is moderate (NMI = 19%).

Now consider a second scenario in which Model is also connected to DurabilityIssue. Mileage is still an influence, but the problem primarily exists for model S280. We change the CPT for node DurabilityIssue to contain $P(\text{yes}|\text{high}, \text{S280})=0.1$, $P(\text{no}|\text{high}, \text{S280})=0.9$, $P(\text{yes}|\text{else}) = 0.01$, and $P(\text{no}|\text{else}) = 0.99$ and simulate 10000 cases. Now, the algorithm first identifies the best partition in the model taxonomy which is $\{\text{S280}\}\{\text{S230}\}\{\text{S350}\}\{\text{W}\}\{\text{C}\}$. From this, it derives the best binary partition Model_H $\{\text{S280}\}\{\text{S230}, \text{S350}, \text{W}, \text{C}\}$ which is also the best partition among the other binary partitions (Figure 6(b)). The partition matrix recommends that the engineer should have a look at the interaction effect of Mileage and Model.H. High mileage *and* a specific group of models (S280) are highly relevant to explain the issue.

Partition	I_B
Mileage	65.3
RentalCar	14.6
Model.H	≤ 0

(a) Scenario 1

	B	
A		RentalCar
	Mileage	A
		19%

Partition	I_B
Model.H	25.0
Mileage	9.4
RentalCar	≤ 0

(b) Scenario 2

	B	
A		Mileage
	Model.H	A
		0%

Figure 6: Ranking of partitions and partition matrix for both scenarios using $I_B(g)$ [PPM(K2)].

4 Evaluation

As we make use of structural information like taxonomies and paronomies and as we restrict the search depth by limiting the maximum number of subsets of a partition, runtime is not critical. In all our experiments, runtime was not longer than half a minute on a standard notebook with 2 GHz and 2 GB RAM. Even for an interactive approach this is sufficient. How counting can be done efficiently for the combination of attributes without minimum support constraint is described in more detail in [3].

Instead of assessing algorithmic performance we rather want to evaluate how interactive look-ahead decision trees based on the application of Bayesian partition models can perform in practice to help engineers in their task of identifying the root cause of quality issues. We test our approach on a publicly available dataset. In addition, we demonstrate the overall value of the system in two real-world case studies.

4.1 Evaluation on Test Data Our first evaluation is based on the credit-scoring dataset described in [9] and [15]. Credit-scoring aims at learning a model that helps in identifying whether a customer is deemed credit-worthy. Although this is rather a classification or prediction task, it shows how our approach can be applied to rank partitions and combinations of partitions. In this example, a “non-conforming” customer is one that is not considered credit-worthy. We increase the skewedness of the target distribution by weighting credit-worthy customers 21 times higher than in the original dataset. Our new dataset contains 15000 customers in total with a portion of 2% of customers that are not credit-worthy.

A uni-variate ranking of the best binary partitions (CurrentAccount CA, PrevPayment PP, Savings S, DurationCredit DC, PurposeCredit PC, TypeApart-

	Partition	I_1	I_2	I_3
CA	{1, 2}{3, 4}	77.9	78.4	70.9
PP	{0, 1}{2, 3, 4}	25.6	24.7	18.7
S	{1, 2}{3, 4, 5}	20.2	20.8	13.3
DC	{> 24}{≤ 24}	15.8	15.7	8.9
PC	{0, 2, 4, 5, 6, 9, 10} {1, 3, 8}	14.8	15.3	7.9
TA	{1, 3}{2}	8.0	8.1	1.1
A	{≤ 25}{> 25}	6.7	6.6	≤ 0
VA	{4}{1, 2, 3}	6.5	6.2	≤ 0
AC	{> 5000}{≤ 5000}	5.5	5.3	≤ 0
TE	{3, 4, 5}{1, 2}	4.8	4.8	≤ 0

Table 1: Ranking of the best binary partitions of the credit-scoring dataset when applying various interestingness measures.

ment TA, Age A, ValueAssets VA, AmountCredit AC, TimeEmployment TE) is given in Table 1. We apply the following interestingness measures to rank partitions: $I_1 = I_B(g)$ [PPM(K2)], $I_2 = I_B(g)$ [Schwarz crit.], and $I_3 = I_O(g)$ [PPM(K2) ($\alpha = 0.001$)]. From the experiments it follows that the ranking is approximately the same for PPM(K2) and the Schwarz criterion. Introducing prior cohesions to penalize larger partitions reduces the prior odds in favor of a partition g and make the prior partition g_- that states that a variable does not have an influence at all, more probable.

The partition matrices for the selected interestingness measures $I_1 - I_3$ are depicted in Table 2. Let us focus on the matrix for I_1 first. We can observe that this is not a typical dataset that contains one or two main influences: CA is the strongest influence, but it is amplified by PP, S, DC, and PC separately. In [15] a CART analysis of this dataset produces a tree with four terminal nodes. This tree corresponds to the interaction of the binary partitions $CA \times PP$ and shows the following: The partition matrix is a compact representation and ranking of the information contained in many decision trees with depth ≤ 2 . The reader might be wondering why the interactions with PP are denoted with $B+$, although PP is ranked higher in the uni-variate ranking. The reason is that PP is a very skewed partition with very high lift for the small group of customers with “hesitant payment of previous credits”. The partition S, for instance, is more balanced and shows high lift for the large group of customers with “no or little savings”. If a customer of this group was hesitant when paying back previous payments he is even less credit-worthy.

Now let us compare the matrices I_1 through I_3 . Especially when using posterior odds instead of the Bayes factor, deviations can be observed (Table 2). Using prior cohesions with small α -values penalizes larger partitions. Consider the interaction effect of partitions CA and S. While $\alpha = 1$ prefers $A+$, $\alpha = 0.001$

Partition	I_1		I_2		I_3	
CA						
PP	A+	94.0	A+	93.2	A+	80.2
S	A+	88.4	A+	88.8	\wedge	79.7
DC	A+	91.3	A+	91.1	A+	77.5
PC	\times	81.7	\times	82.7	A	70.9
PP						
S	B+	48.5	B+	47.7	B+	34.7
DC	B+	42.4	B+	41.1	B+	28.6
PC	B+	38.2	B+	37.3	\wedge	25.3
S						
DC	A+	44.8	A+	44.8	\wedge	31.8
PC	\wedge	30.1	\wedge	30.5	\wedge	23.2
DC						
PC	\times	30.0	\times	29.7	\wedge	19.2

Table 2: Best partition combinations for the top 5 variables of the credit-scoring dataset for various partition models.

prefers the simpler \wedge . The reason for this is that the subset of customers with “no running account or debt” that have “no or little savings” is much less credit-worthy. Customers with “no running account or debt”, but with “savings” are also less credit-worthy, but to a smaller degree. Other customers are mainly considered credit-worthy. This illustrates that the analyst can focus on the most important influences by decreasing the cohesions α . On the other hand, increasing α reveals more details, but may lead to overfitting.

Structure learning of Bayesian networks, e.g. the GES procedure [5], works well for the simple simulation studies in the last section. However, in this example a network of strongly connected influential variables is found, but *only* CA is connected to the class variable. On the other hand, association rule mining yields hundreds of interesting subgroups.

4.2 Real-World Case Studies Finally, we want to present two case studies that illustrate how our interactive tool is applied in the context of analyzing warranty data. The system provides our users with a ranked list of partitions that have the strongest influence on an issue and supports an interactive investigation.

Our first example is also discussed in [3]. Several vehicles are brought to dealerships because a lamp indicates an engine issue. Diagnostic tools point to problems with the exhaust system. Finding no trouble, dealers replace the oxygen sensors. Warranty costs for these sensors increase significantly while quality engineers cannot find an explanation for the issue. In particular a thorough check of the replaced sensors and other parts of the exhaust system indicates that neither the sensors nor other parts seem to have failed.

As the engineers know that only one engine type can

set the fault code, we apply a first manual split in the decision tree and restrict the dataset to all instances with `Opt_Engine=E`. Besides, we know that all service claims are related to the CARB states emissions system and restrict the dataset to vehicles with `Opt_Emission=N`.

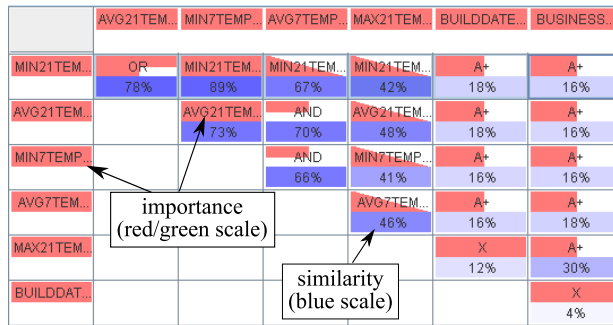


Figure 7: Partition matrix containing the highest ranked influences for the CARB states example [3] after having restricted engine type and emissions standard. Temperature, BuildDate and BusinessCenter show up as primary influences.

Now we calculate the partition matrix. The best influences and their interactions are given in Figure 7. Temperature, for instance the average temperature within the last 21 days before repair date (Min21Temp), BuildDate and BusinessCenter are the strongest influences. High NMI values and the blue background color indicate that all temperature partitions describe similar vehicle subsets. Temperature seems to be the primary influence, but BuildDate and BusinessCenter have an amplifying effect. As we mentioned in [3], the true cause is a calibration issue: The fault code was erroneously set under rare conditions, in particular when the environment temperature was low and the engine was in wide open throttle mode, i.e. when the vehicle was accelerating strongly. As these conditions are more likely to hold, when people drive on highways, an interaction effect of temperature and region makes sense.

The assessment for the interaction of Min21Temp \times BusinessCenter is represented in Figure 8. The strongest partition is A+, i.e. Min21Temp+, but it is followed closely by Min21Temp. There exists an amplifying effect of BusinessCenter, which means that the problem occurs in particular when temperature is low and when BusinessCenter is NorthEast, but in general it is primarily related to cold temperatures. The build date range probably shows up because of software updates for the engine control module.

Our second example is related to “seat occupancy recognition”, a safety-related feature that checks whether passengers properly fastened their seat belts

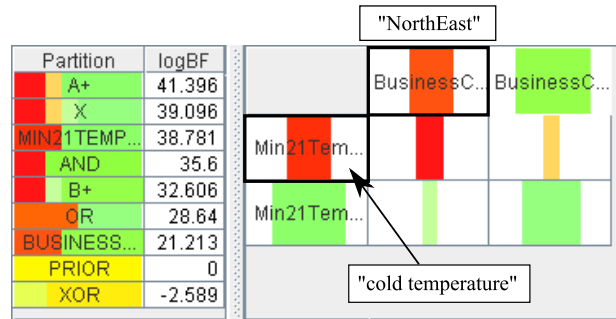


Figure 8: Detail view of the combination of the binary partitions Min21Temp and BusinessCenter.

and that deactivates the passenger airbag if a child seat is installed. A special mat weaved into seats recognizes the weight of a person sitting on it. In case a seat belt is not fastened properly and the vehicle starts to move, this is indicated by an acoustic signal. The problem was that some seats had to be replaced because the mat was damaged.

Again we want to show step-by-step how our interactive data analysis can help the engineer: The interactive decision tree tells the analyst that only a specific model series is affected by the issue and so he can narrow down the dataset. On this dataset that contains about 140 potential influence variables, the partition matrix is generated.

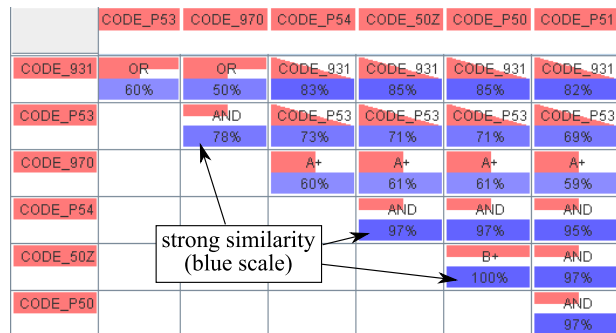


Figure 9: The seat occupancy recognition example illustrates the similarity feature of the partition matrix. Several police related sales codes are ranked highest and refer to the same subset of vehicles.

An excerpt from the highest ranked influences can be seen in Figure 9. The top codes all refer to special equipment for police cars: a box for weapons in the trunk, alert lights on the cover, or integrated radar equipment. All these codes describe a similar subset of vehicles, which is indicated by the blue color and the high NMI similarity values. Note that we would

not recognize this similarity, if we did not look ahead one level. This is a first influence: Policemen wear weapons and other heavy equipment at their belts. Whenever they enter a car quickly, they are likely to cause damage to the mat under the seat cover. However, this explanation only accounts for some of the failures. Not all vehicles are driven by policemen and only one third of all issues are explained by the police influence. Another advantage of our interactive approach is that we can simply create a new analysis dataset by separating all police cars from the other vehicles in the decision tree (for example by deriving a new attribute that is true whenever any of the police equipment is present). Note that this would not be possible in a fully automatic setting.

On the new dataset, the partition matrix is generated again. Now vehicles shipped to Japan and Australia are especially likely to be affected by the issue. A specific production plant that primarily produces vehicles for these regions, a production period and the steering type show up, too. In a uni-variate setting, all influences would seem equally likely, but the partition matrix reveals that the latter two have to be considered non-causal. Hence, the quality engineer should primarily investigate whether the issue could be related to the shipping of vehicles.

5 Discussion of Related Work

The subgroup mining approach [17] has gained notable attention because of efficient search strategies that can enumerate all subgroups fulfilling certain constraints. However, a major issue is the huge number of interesting subgroups: Although subgroups can be ranked, clustered and non-causal subgroups can be suppressed, the remaining number of interesting subgroups can be tremendous. Moreover, in our context the subgroup focus proves to be quite restrictive: While investigating an issue, the user wants to interactively explore the neighborhood of a subgroup. Partition models are an extension to subgroup discovery as a partition is a grouping of related subgroups that hides the uninteresting ones.

Relaxing the subgroup focus is also the main advantage of interactive rule cubes as proposed in [27] and [3]. Apart from that, rule cubes support an interactive causal validation by suggesting possibly non-causal findings and by allowing users to suppress expected influences. However, the interestingness of a cube is defined by a single cube cell. This is critical when the domain gets large and attribute values are not grouped. Another issue is that although the interestingness measure optimizes the trade-off between precision and recall, it cannot be interpreted in statistical terms.

Interactive decision trees are yet another implemen-

tation of subgroup discovery or rather *subgroup description* [2]. Due to their intuitive representation, interactive decision trees have gained notable acceptance among quality engineers. Interactivity plays a key role in causal investigations: When splitting a node, an engineer might not accept the recommended variable or variable grouping, but he will adjust the split according to his background knowledge. The observation that only a limited set of hypotheses can be explored interactively and that identifying similar or non-causal findings is not straightforward lead to the development of interactive look-ahead decision trees.

Learning (causal) Bayesian networks has been considered the matter of choice for investigating cause-effect relationships for years. Various Bayesian and constraint-based approaches have been developed to learn the structure of (causal) Bayesian networks [7, 12, 24, 20, 5]. In theory the computational complexity of this task is challenging. In practice, explaining a learned network structure and concepts like d-separation to an engineer is even more difficult. In contrast, a partition representing subsets of vehicles is rather intuitive. And still we have to be aware of causality if we want to reduce the risk of wrong decisions. While Bayesian networks represent the *global* dependence structure of the data, in our application context one rather has to focus on the *local* neighborhood of a single (target) variable. Furthermore, Bayesian networks are learned incrementally and a resulting network might be meaningless if variables that are not relevant for an analysis (artifacts) are included at an early stage. An interactive analysis that incorporates users' feedback is more robust. Semi-automatic approaches that allow a user-driven, iterative construction of Bayesian networks seem to be a very promising alternative in our application context [13].

6 Conclusion and Future Work

In this paper, we show how Bayesian partition models can be applied in an interactive approach to support root cause investigations in the automotive industry. We decompose the specifics of this application and show in a detailed evaluation how our approach is suited to handle these issues. Our approach extends the work on interactive decision trees [2] and rule cubes [3] and is mainly based on research done in the field of Bayesian partition models [11, 1, 6, 10]. The key idea is that actionable knowledge can only be derived from semantically meaningful partitions. A set of reasonable assumptions reduces the partition space that otherwise could only be explored by applying MCMC methods. In addition, we exploit semantic hierarchies to create meaningful value groupings for categorical variables and search the most likely partition on a numeric

domain. The partition matrix visualizes interactions of binary partitions and highlights non-causal and similar partitions. Ranking and comparing of partitions is done by applying ideas from Bayesian model comparison. Interactive look-ahead decision trees are an extension to regular interactive decision trees and make it easier for a user to choose the best split attribute. The system highlights non-causal or similar attributes and recommends a split by looking ahead two levels.

There are several directions for future research. First, as pointed out by an anonymous reviewer, Bayesian network classifiers should be considered: In fact, e.g., *Tree Augmented Naive Bayes* (TAN) could be a promising alternative as taxonomies can be incorporated and the structure can be learned efficiently. The expressive power of TANs and other Bayesian network classifiers should be compared to our approach. Second, we want to extend the approach to support numeric target attributes like repair costs. Last but not least, we want to address a seamless integration with OLAP reporting tools, as the approach proves to be especially valuable when semantic hierarchies exist.

References

- [1] D. Barry and J. A. Hartigan, *Product partition models for change point problems*, The Annals of Statistics, 20 (1992), pp. 260–279.
- [2] A. Blumenstock, J. Hipp, S. Kempe, C. Lanquillon, and R. Wirth, *Interactivity closes the gap*, in Proceedings of the KDD Workshop on Data Mining for Business Applications, Philadelphia, USA, 2006.
- [3] A. Blumenstock, F. Schweiggert, and M. Mueller, *Rule cubes for causal investigations*, in Proceedings of the ICDM, Omaha, NE, USA, 2007, pp. 53–62.
- [4] C. G. Borroni and R. Piccarreta, *Genetic algorithms for the analysis of bayesian hierarchical partition models*, Statistical Methods and Applications, 10 (2001), pp. 113–121.
- [5] D. M. Chickering, *Optimal structure identification with greedy search*, Journal of Machine Learning Research, 3 (2002), pp. 507–554.
- [6] G. Consonni and P. Veronese, *A bayesian method for combining results from several binomial experiments*, Journal of the American Statistical Association, 90 (1995), pp. 935–944.
- [7] G. F. Cooper and E. Herskovits, *A bayesian method for the induction of probabilistic networks from data*, Machine Learning, 9 (1992), pp. 309–347.
- [8] E. M. Crowley, *Product partition models for normal means*, Journal of the American Statistical Association, 92 (1997), pp. 192–198.
- [9] L. Fahrmeir and G. Tutz, *Multivariate Statistical Modelling Based on Generalized Linear Models*, Springer, New York, 1994.
- [10] D. Golinelli, D. Madigan, and G. Consonni, *Relaxing the local independence assumption for quantitative learning in acyclic directed graphical models through hierarchical partition models*, in Seventh International Workshop on Artificial Intelligence and Statistics, 1999, pp. 203–208.
- [11] J. A. Hartigan, *Partition models*, Communications in Statistics, Part A - Theory and Methods, 19 (1990), pp. 2745–2756.
- [12] D. Heckerman, D. Geiger, and D. M. Chickering, *Learning bayesian networks: The combination of knowledge and statistical data*, Machine Learning, 20 (1995), pp. 197–243.
- [13] S. Jaroszewicz and D. A. Simovici, *Interestingness of frequent itemsets using bayesian networks as background knowledge*, in Proceedings of KDD, New York, NY, USA, 2004, ACM Press, pp. 178–186.
- [14] H. Jeffreys, *Theory of Probability*, Oxford University Press, Oxford, U.K., 3 ed., 1961.
- [15] C. Jordan, V. Livingstone, and D. Barry, *Statistical modelling using product partition models*, Statistical Modelling, 7 (2007), pp. 275–295.
- [16] R. E. Kass and A. E. Raftery, *Bayes factors*, Journal of the American Statistical Association, 90 (1995), pp. 773–795.
- [17] W. Klösgen, *Advances in subgroup mining*. Fraunhofer Institute for Autonomous Intelligent Systems, Sankt Augustin, Germany, 2004.
- [18] M. Koivisto and K. Sood, *Computational aspects of bayesian partition models*, in Proceedings of ICML, Bonn, Germany, 2005.
- [19] D. J. MacKay, *Bayesian Methods for Adaptive Models*, PhD thesis, California Institute of Technology, 1991.
- [20] J. Pearl, *Causality: Models, Reasoning, and Inference*, Cambridge University Press, 2000.
- [21] CMU Tetrad Project, *The tetrad project: Causal models and statistical data*. <http://www.phil.cmu.edu/projects/tetrad/>, 2008.
- [22] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, 1993.
- [23] F. A. Quintana and P. L. Iglesias, *Bayesian clustering and product partition models*, Journal of the Royal Statistical Society, 65 (2003), pp. 557–574.
- [24] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, MIT Press, 2000.
- [25] H. Steck, *Learning the bayesian network structure: Dirichlet prior versus data*, in Proceedings of UAI, 2008.
- [26] A. Strehl and J. Gosh, *Cluster ensembles - a knowledge reuse framework for combining multiple partitions*, Journal of Machine Learning Research, 3 (2002), pp. 583–617.
- [27] K. Zhao, B. Liu, J. Benkler, and W. Xiao, *Opportunity map: Identifying causes of failure - a deployed data mining system*, in Proceedings of KDD, New York, NY, USA, 2006, ACM, pp. 892–901.