BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 58, No. 3, 2010
DOI: 10.2478/v10175-010-0037-z

Visual data analysis with computational intelligence methods

R. KRUSE* and M. STEINBRECHER

Department of Knowledge and Language Engineering, Faculty of Computer Science
Otto-von-Guericke-University of Magdeburg, D-39016 Magdeburg, Germany

Abstract. Visual data analysis is an appealing and increasing field of application. We present two related visual analysis approaches that allow for the visualization of graphical model parameters and time-dependent association rules. When the graphical model is defined over purely nominal attributes, its local structure can be interpreted as an association rule. Such association rules comprise one of the most prominent and wide-spread analysis techniques for pattern detection, however, there are only few visualization methods. We introduce an alternative visual representation that also incorporates time since patterns are likely to change over time when the underlying data was collected from real-world processes. We apply the technique to both an artificial and a complex real-life dataset and show that the combined automatic and visual approach gives more and faster insight into the data than a fully-automatic approach only. Thus, our proposed method is capable of reducing considerably the analysis time.

Key words: visual data analysis, computational intelligence methods.

1. Introduction

Data analysis is a vital component in strategic planning for companies that are aware of global competition, ever-shorter production cycles and increasing customer requirements. It is of paramount importance to identify meaningful patterns quickly within the collected data in order to respond to impending supply shortages or evolving problems with delivered products. However, patterns that correspond to such lingering problems rarely occur out of a sudden. Therefore we have developed a temporal view on the data as well as the resulting patterns. Further, we intend to enable users that not necessarily have a statistical background to assess and understand the identified patterns. This has been accomplished by devising appropriate visualization methods for patterns as well as their temporal change. The setting within which our research, development and application took place is the automobile manufacturing industry. Especially under the more stringent constraints imposed by the worldwide financial crisis, it is of paramount importance to respond to any potential problem related to vehicle safety or quality before a widespread recall is inevitable that will come at an enormous expense and loss of customer confidence.

The modeling technique used for solving the outlined issues shall accommodate two main aspects: firstly, it must allow for a global view on the domain that is under analysis, i.e. the overall interconnections and interrelations between the attributes that describe a vehicle. As these are normally high-dimensional, a compact but still usable knowledge representation has to be found. Secondly, the user must be enabled to inspect any local dependency in greater details if he wishes to.

To illustrate these two claims in the realm of a vehicle manufacturer, assume that every vehicle configuration is stored in a database. Such a configuration often contains several tens to sometimes hundreds of attributes and hence dimensions. The stochastic dependencies – and more important: independencies – will be represented by a (directed) graph in which a node models an attribute amongst which the dependencies are reflected by edges. In our application this graph will be created from the database with optional preceding or subsequent expert-specified alterations. This will allow the user (e.g. an engineer or marketing analyst) to infer coarse-grained conclusions based on the potential effects between connected attributes. When it comes to a question that is narrowed down to a specific configuration fragment, the parameters attached to every node in the graph can reveal answers to quantitative questions such as “Whenever a repair report referenced transmission type X, there is a 40% chance of also having the engine type Y built into the car, which rises the failure rate by 30%”. In this case dependencies are contained in the vehicle database and are not known beforehand but are extracted to reveal possible hidden design flaws. This example calls for treatment methods that exploit the dependence structures embedded inside the application domains. We chose graphical models, more specific: Bayesian networks, to address these issues. The two abovementioned criteria map well onto the global and local components a graphical model consists of.

Until now, we neglected the discussion of one important dimension: time. For obvious reasons, a vehicle manufacturer wants to counteract problems before they seriously affect huge number of cars. That is, if a problem with a rust-prone part in the suspension is discovered, then an early finding of this can lead to the advice to exchange this part during the next regular checkup. If done early enough it will avoid the need of a possible recall when the majority of cars will be eventually affected having exceeded a certain mileage threshold.

*e-mail: kruse@iws.cs.uni-magdeburg.de

Roughly speaking, patterns normally do not arise all of a sudden but evolve over time. Thus, the number of failed cars (and thus possibly interesting failure patterns) grows larger slowly. In contrast to this, some countermeasures undertaken will take some time to have an apparent effect, thus rendering the decrease of the failure pattern slowly as well.

We augmented the outlined analysis tool with a means of filtering temporal patterns. As will be shown later, any parameter attached to a node in a graphical model can be seen as an association rule [1]. This has proven to be very fortunate since this type of model is widely understood and thus accepted amongst users. In addition to that, any rule mining technique can be used to postprocess the identified rules, if wanted.

Our postprocessing step for rules will rely on linguistic expressions in terms of a fuzzy description which constrains the temporal behavior or evolution of a rule. By evolution in time we refer to the change of certain rule evaluation measures that we will discuss later on.

The next section will briefly introduce the needed background on graphical models and association rules. Section 3 presents the application of visualizing parameters of graphical models after which Sec. 4 discusses how to extend the approach to accommodate for temporal change in the patterns. Both sections, of course, include examples. Section 5 concludes the article.

2. Background

We will now briefly discuss the notational underpinning that is needed to present the ideas and results from the industrial applications.

2.1. Graphical models. As we have pointed out in the introduction, there are dependencies and independencies that have to be taken into account when reasoning in complex domains shall be successful. Graphical models are appealing since they provide a framework of modeling independencies between attributes and influence variables. The term "graphical model" is derived from an analogy between stochastic independence and node separation in graphs. Let $V = \{A_1, \ldots, A_n\}$ be a set of random variables. If the underlying probability distribution $P(V)$ satisfies some criteria (see e.g. [2, 3]), then it is possible to capture some of the independence relations between the variables in V using a graph $G = (V, E)$, where E denotes the set of edges. The underlying idea is to decompose the joint distribution $P(V)$ into lower-dimensional marginal or conditional distributions from which the original distribution can be reconstructed with no or at least as few errors as possible [4, 5]. The named independence relations allow for a simplification of these factor distributions. We claim, that every independence that can be read from a graph also holds in the corresponding joint distribution. The graph is then called an independence map.

If we are dealing with an acyclic and directed graph structure G, the network is referred to as a Bayesian network. The decomposition described by the graph consists of a set of conditional distributions assigned to each node given its direct predecessors (parents). For each value of the attribute domains (dom), the original distribution can be reconstructed as follows:

$$\begin{aligned}&\forall a_1 \in \mathrm{dom}(A_1) : \cdots \forall a_n \in \mathrm{dom}(A_n) :\\ &\quad P(A_1 = a_1, \ldots, A_n = a_n) =\\ &\qquad\qquad \prod_{A_i \in V} P\Big(A_i = a_i \mid \bigwedge_{(A_j, A_i) \in E} A_j = a_j\Big)\end{aligned}$$

2.2. Association rules. The introduction of frequent item set mining and subsequently association rule induction [1, 6] has created a prospering field of data mining. It is the simplicity of the underlying concept that allowed for a broad acceptance among all kinds of users no matter whether they possess a data analysis background or not. An association rule is basically an *if-then* rule. The *if*-part is called *antecedent* while the *then*-part is named the *consequent*. Both may consist of conjunctions of attribute-value pairs, however, the consequent often consists of only one pair. An example of an association rule could be

> If a person is male and a smoker, his probability of having lung cancer is 10%.

This corresponds to the imagination that we pick a person at random from an underlying population (the database) and observe its properties, that is its attribute values. The above rule can then be represented in a more formal fashion as

$$\text{Gender} = \text{male} \ \wedge \ \text{Smoker} = \text{yes} \ \rightarrow \ \text{Cancer} = \text{yes}. \quad (1)$$

We refer to a database case as being covered by a rule if the antecedent and consequent attributes values match. For instance, a smoking man having lung cancer would be covered by the above rule. The general form of a rule has the following form:

$$A_1 = a_1 \wedge \cdots \wedge A_n = a_n \longrightarrow C = c \quad \overset{\text{abbr}}{=} \quad \vec{a} \rightarrow c$$

We will only discuss rules with one consequent attribute which will be a class variable. We thus use the notions class and consequent interchangeably.

Since not every database entry matching the antecedent also matches the consequent it is necessary to record this information. The probability that a database case matching the antecedent also matches the consequent, that is $P(c \mid \vec{a})$, is called the *confidence* of the rule. The above rule 1 has a confidence of 0.1. There is a multitude of other measures that quantify certain aspects of a rule (see, e.g. [7]). We will briefly discuss those that are used in this paper.

The number of cases covered by the rule is referred to as the (absolute) *support* of the rule. The relative support equals $P(\vec{a}, c)$; it is the absolute support divided by the database size. The *recall* quantifies the fraction (or probability if you keep the above scenario of picking at random) of database cases matching the antecedent, given the consequent. In other words: What is the probability of a person being male and a smoker if this person has cancer? As a last measure (the only unbounded one) we introduce the *lift*. It represents the ratio between the confidence $P(c \mid \vec{a})$ and the marginal consequent probability $P(c)$: Let the marginal cancer rate be 0.01. Then, rule 1 has a lift of 10 since the confidence is ten times larger than the marginal cancer rate. We summarize the measures below:

– relative support: $\text{rel-supp}(\vec{a} \to c) = P(\vec{a}, c)$

– confidence: $\text{conf}(\vec{a} \to c) = P(c \mid \vec{a})$

– recall: $\text{recall}(\vec{a} \to c) = P(\vec{a} \mid c)$

– lift: $\text{lift}(\vec{a} \to c) = \dfrac{P(c \mid \vec{a})}{P(c)}$

3. Fault analysis with graphical models

For every car that is sold, a variety of data is collected and stored in corporate-wide databases. After every repair or check-up the respective records are updated to reflect the technical treatment. The analysis scenario discussed here is the interest of the automobile manufacturer to investigate car failures by identifying common properties that are exposed by specific subsets of cars that have a higher failure rate.

3.1. Data description and model induction. As stated above, the source of information consists of a database that contains for every car a set of several tens or hundreds of attributes that describe the configuration of every car that has been sold.

The decision was made to use Bayesian networks to model the dependence structure between these attributes to be able to reveal possible interactions of vehicle components that cause higher failure rates. The induction of a Bayesian network consists of identifying a good candidate graph that encodes the independencies in the database. The goodness of fit is estimated by an evaluation measure. Therefore, usual learning algorithms consist of two parts: a search method and the mentioned evaluation measure which may guide the search. Examples for both parts are studied in [8–10].

Given a network structure, an expert user will gain first insights into the corresponding application domain. In Fig. 1 one could identify the mileage to have a major (stochastic) impact on the failure rate and type. Of course, arriving at such a model is not always a straightforward task since the available database may lack some entries requiring the treatment of missing values. In this case possibilistic networks [11] may be used. However, with full information it might still be problematic to extract significant statistics since there may be value combinations that occur too scarcely. Figure 2 shows a real-world network structure that was induced from a given database. An expert can already benefit from the encoded stochastic direct and indirect dependencies in order to come up with hypotheses what attributes might be most predictive w.r.t. the failure attribute. However, the bare network structure does not reveal information about which *which* mileages have *what kind* of impact on *which* type of failure. Fortunately, this information can be retrieved easily in form of conditional probabilities from the underlying dataset, given the network structure. This becomes clear, if the sentence above is restated: Given a specific mileage, what is the failure probability of a randomly picked vehicle?

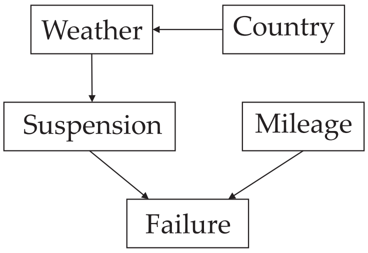

Fig. 1. The qualitative component of an exemplary Bayesian network

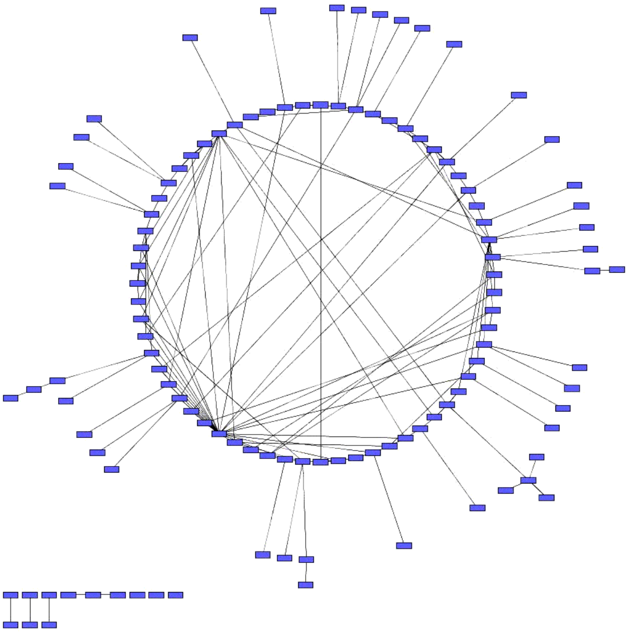

Fig. 2. The qualitative component of a real-world Bayesian network

3.2. Model visualization. Every attribute together with its direct parent attributes encodes a set of conditional probability distributions. For example, given a database D, the sub-network consisting of Failure, Suspension and Mileage in Fig. 1 defines the following set of distributions:

$$P_D(\text{Failure} \mid \text{Suspension}, \text{Mileage})$$

For every distinct combination of values of the attributes Suspension and Mileage, the conditional probability of the attribute Failure is estimated (counted) from the database D. As every such distribution is one-dimensional in the argument (only one attribute, namely Failure in contrast to possibly multiple attributes in the condition), we can depict the failure node's distributions with small number of parent attributes as shown in the two examples of Fig. 3: Every distribution of Failure (which here has two values: yes or no), given a distinct combination of values of Suspension and Mileage is represented as a stacked bar chart. The dark region corresponds to the case Failure=yes. The width of the stacks encode the probability of the condition, i.e. a measure for the number of

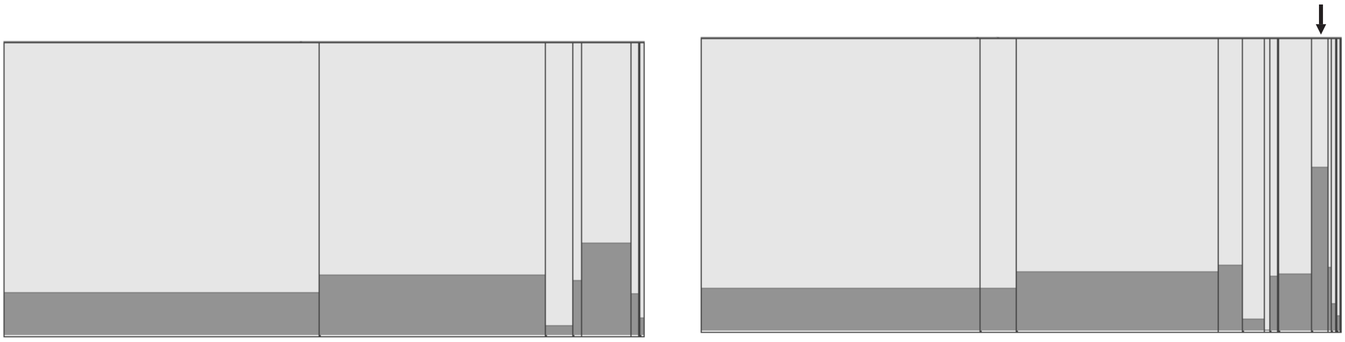

Fig. 3. Left: Conditional distributions of the failure node when only one dependent attribute is present. Right: Conditional distributions with two attributes selected as condition. The peak marked with the arrow reveals a much higher failure rate for this condition than for the others. Figure fig.kg-eg shows an alternative representation of the same distributions

cars that actually belong to that condition configuration. The left of Fig. 3 shows only one conditional attribute which has apparently 7 values resulting in 7 conditional distributions of the failure attribute. There is a conditional dependence between the condition attribute and the failure attribute as the distributions clearly differ. A true interesting aspect will not be visible until a second attribute is selected into the condition leading to the left chart. The peak failure rate (marked with the arrow) for one specific subset of cars is now visible and can be elected subject to further investigation.

Given an attribute of interest (in most cases the class variable like Failure in the example setting) and its conditioning parents, every probability statement like

$$P(\mathsf{Failure} = \mathsf{Bearings\,broken} \mid \mathsf{Suspension} = \mathsf{Type\ X}, \mathsf{Mileage} = \mathsf{over\ 100K\ mi}) = p^*$$

can be considered an association rule:

> If Suspension = Type X ∧ Mileage = over 100K mi, then there will be a bearings failure in $100 \cdot p^*\%$ of all cases.

The value p^* is then the confidence of the corresponding association rule (c.f. Sec. 2). Of course, all known evaluation measures can be applied to assess the rules. With the help of such measures one can create an intuitive visual representation according to the following steps:

- For every probabilistic entry (i.e., for every rule) of the considered conditional distribution $P(C \mid A_1, \ldots, A_m)$ a circle is generated to be placed inside a two-dimensional chart.
- The gray level (or color in the real application) of the circle corresponds to the value of attribute C.
- The circle's area corresponds to the value of some rule evaluation measure selected before displaying. For the remainder of this chapter, we choose this measure to be the support, i.e., the relative number of vehicles (or whatever instances) specified by the values of C and $A_1, \ldots, A_m$. Therefore, the area of the circle corresponds to the number of vehicles.
- In the last step these circles are positioned. Again, the value of the x- and y-coordinate are determined by two evaluation measures selected in advance. We suggest these measures to be confidence and lift. Circles above the darker horizontal line in every chart mark subsets with a lift greater than 1 and thus indicate that the failure probability is larger given the instantiation of $A_1, \ldots, A_n$ in contrast to the marginal failure probability $P(C = c)$.

With these prerequisites we can issue the user the following heuristic in order to identify suspicious subsets:

> Sets of instances in the upper right hand side of the chart may be good candidates for a closer inspection.

The greater the y-coordinate (i.e. the lift value) of a rule, the stronger is the impact of the conditioning attributes' values on the class variable. Larger x-coordinates correspond to higher confidence values.

3.3. Application. This section illustrates the proposed visualization method by means of three real-world datasets that were analyzed during a cooperate research project with a automobile manufacturer. We used the K2 algorithm [8] to induce the network structure and visualized the class variable according to the given procedure.

1) Example 1. Figure 4 shows the analysis result of approximately 60000 vehicles. Attributes Precipitation and Transmission had most (stochastic) impact on the Failure variable. The subset marked by the arrow was re-identified by experts as a problem already known.

2) Example 2. The second dataset consisted of approximately 200000 cars that exposed a many-valued class variable, hence the different gray levels of the circles in Fig. 5. Although there was no explanation for the subset 3, the other two could be tracked back to known dependencies of the respective values of the conditioning attributes.

3) User acceptance. The proposed visualization technique has proven to be a valuable tool that facilitates the identification of subsets of cars that may expose a critical dependence between configuration and failure type. Generally, it represents an intuitive way of displaying high-dimensional, nominal data. A pure association rule analysis needs heavy postprocessing of the rules since due to the commonly small

failure rate a lot of rules are generated. The presented approach can be considered a visual exploration aid for association rules. However, one has to admit, that the rules represented by the circles share the same attributes in the antecedence, hence the sets of cars covered by these rules are mutual disjoint, which is a considerable difference to general rule sets.

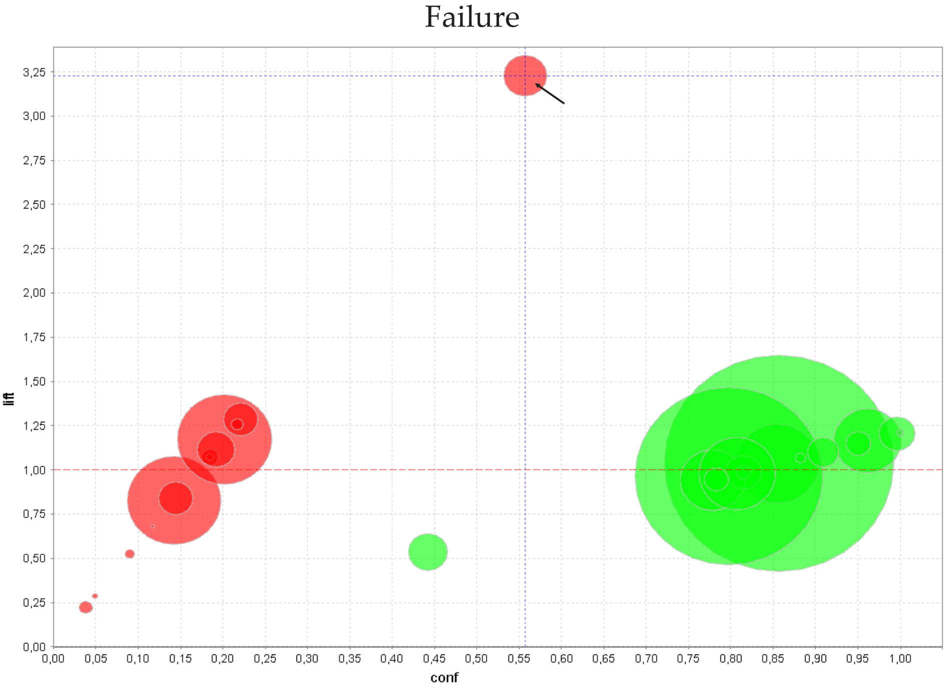

Fig. 4. The subset marked by the arrow corresponds to 825 vehicles whose attributes values of Precipitation and Transmission yielded a causal relationship with the class variable

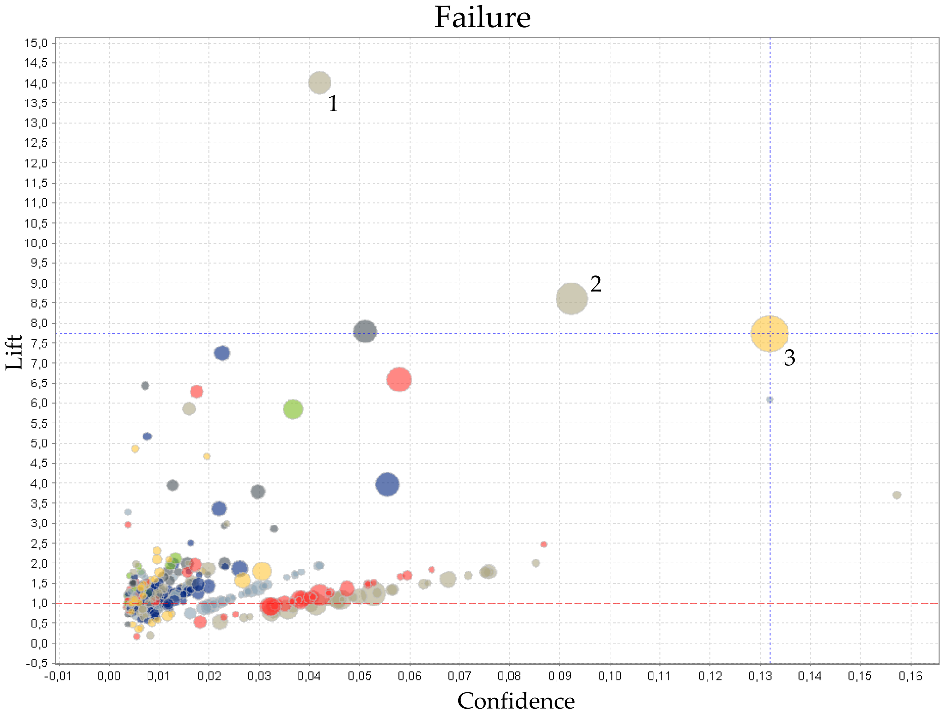

Fig. 5. The three numbered subsets represent 607, 1231, and 1759 cars each. Subset 1 and 2 belong to the same class and differ only in the condition. It now depends on the intention of the expert analyzing such charts which subsets to address more attention: if he is interested in the relative confidence increase only, he would go for subset 1 since the lift has a value of 14. That is, given the condition the failure probability is raised to 4.2% which is 14 times the apriori failure rate of 0.4%. If the user is interested in maximum rule validity instead, he would investigate rule 2 since it has a higher confidence of 9.2%

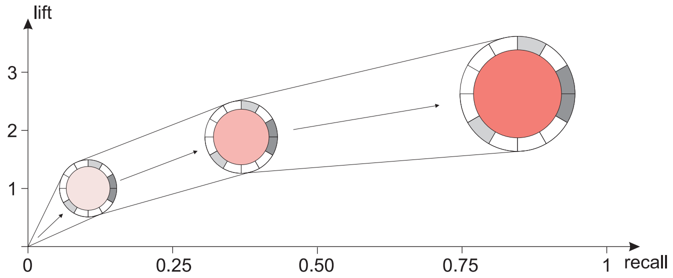

Fig. 6. A rule at three different times. Initially, the rule did not cover any database case, that is its initial icon is just a point at the origin. As time passes, the support increased (growing size of the circle), the recall and lift did alike (circle is moving to the upper right-hand corner). The user would be presented an animation with a smooth transition between the three states

4. Temporal change in graphical models.

As motivated in the introduction, we now develop the analysis of patterns encoded in a graphical model further and also address the temporal aspect which will turn out to be a post-processing step to a set of rules [12–15]. The temporal evolution of such patterns carries valuable – if not vital – information about the urgency of the underlying problem (or the effectiveness of the treatment). Again, we intend to involve the human user into the analysis process as a fully automatic approach has its limitations. In order to minimize response times to problems, data analysis results must be interpretable by technical staff that normally has no statistical background. In addition, the analysis should be as transparent as possible to comprehend all inferences and conclusions that were drawn.

We first enhance the rule visualizaton technique from Subsec. 3.2 and then augment it to cater for the presentation of temporal change of these rules (or more specific the change of rule properties).

4.1. Temporal pattern visualization. In this section we first introduce the visualization of rule sets without respect to time.

After having established the intuition for this method, we apply the visualization to rule sets from different time frames. Examples are presented and discussed in Subsec. 4.6.

4.2. Extended rule visualization. In addition to the rule visualization of Subsec. 3.2 where a rule was depicted by a colored circle, we now employ a more sophisticated icon. Such an icon is depicted in Fig. 8: Still, every rule is represented as a circle the size of which represents the support of the rule.

The interior is solidly filled with a color denoting the consequent attribute value. The saturation of this color is used to indicate the confidence of the rule: Full saturation represents 100% confidence (that is a deterministic rule) while white would technically correspond to 0% validity. One may argue that rules with different consequent attribute values and thus different interior colors might be hard to distinguish if they both have a low confidence. This is correct, however, rules with low confidence are not of interest and will not be generated by a respective rule induction algorithm (since a minimum confidence acts as a threshold). In addition to that, in practical cases, we often observed the user to be only interested in one designated class value, neglecting all others. In such a case, there will be only one color tone (for example red) with different saturations.

The antecedent of the rule is shown in the border of the circle. For every possible attribute (i.e. for every attribute except the class attribute) a unique fragment is reserved. The fragments are equally sized and are ordered clockwise, starting at the right. We did not use the additional degree of freedom given by the potential different sizes of the fragments for further quantitative encodings since the icon already contains a multitude of information. Comparing the sizes of respective fragments of two rules would be hard because it scales with the rule support. A third counter-argument is that since all fragments form a circle a change of the size of one fragment would change all others, too.

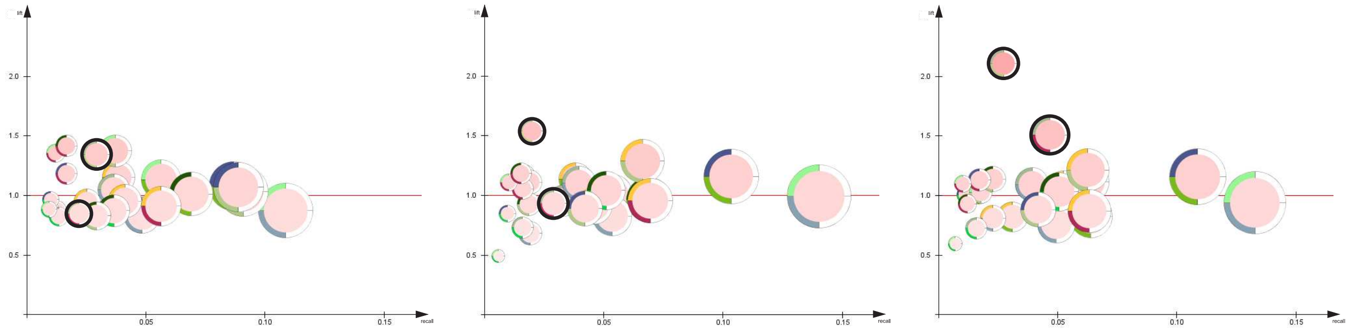

Fig. 7. The 25 rules evaluated w.r.t.the artificial car manufacturer dataset discussed in Sec. 4.5. The antecedence of every rule represents a unique combination of air conditioning type and country. Only rules with consequent failure=yes are shown. The charts show the rule set at the three times Jan, Feb and Mar. Clearly, two rules exhibit a faster movement that indicates an increasing lift. For better assessment, the trajectories of these two rules have been repeated in Fig. 11

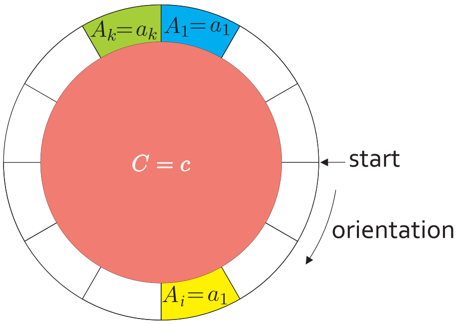

Fig. 8. The visualization of a single association rule as it is used in this paper. The outer ring encodes the values of the antecendet attributes whereas the interior represents the class value and the confidence of the rule

The order of the fragments, that is the order of the represented attributes is free of choice as long as it is the same for all rules. We ordered the attributes alphabetically thus making the order independent from the data.

Every fragment is filled according the following policy: If the respective attribute is referenced in the antecedent, the corresponding fragment is filled with a color that uniquely represents the value of the attribute's domain. This way of representation, of course, is only feasible if the number of attributes and the size of their domains is small. Otherwise, we simply omit the other antecedent ring. However, in the real-world data used to evaluate the method to be proposed in this section, the underlying domain allowed for a representation as described above.

The arrangement in the chart follows the same rules as in Subsec. 3.2 with only one difference: We assign as coordinates the value of association rule evaluation measures. The lift value of a rule is used as its y-coordinate, however, we now use the recall as its x-coordinate. The confidence as a third interesting measure is represented by the (de-)saturated inner color and the support is represented by the circle area. Doing so, we are able to encode four numeric dimensions into a two-dimensional image without redundancy.

4.3. Overlapping rules. Up to now, we assumed the rules to cover mutual disjoint sets of database objects, i.e., every entry of the database was described by exactly one rule antecedent. This can be easily achieved by requesting a fixed set of attributes for every rule. If the user, however, is interested in general rules where database entries may be covered by multiple rules (e.g. because one rule is a specialization of another), we have to cater for this fact by depicting the mutual overlap.

Consider a population for which we assess the probability of having lung cancer. Let the cancer probability for a male person be 15%, i.e., the rule

$$\text{Gender} = \text{male} \rightarrow \text{Cancer} = \text{yes}$$

has a confidence of 0.15. Let this confidence increase to 30% when the additional information that the person is a smoker is known. The respective rule is

$$\text{Gender} = \text{male} \wedge \text{Smoker} = \text{yes} \rightarrow \text{Cancer} = \text{yes}.$$

Clearly, all persons covered by the antecedent of the second rule are also covered by the antecedent of the first rule, hence they cover non-disjoint sets of cases. To depict this, we connect rule visualization by a line in a chart whenever they share a set of common objects covered by both antecedents. Further, we compare the cardinality of this intersection to the support of both rules. The two ratios between intersection cardinality and the two rule supports are indicated as a bar chart on that connecting line. The 100%-mark is located in the middle, whereas the 0%-mark is on the rules' outer border. Figure 9 depicts the example situation of lung cancer above. The used numbers of cases are given in Table 1 for the sake of completeness.

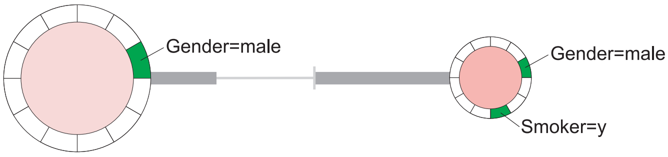

Fig. 9. Visualizing the overlap of two rules. Since "male∧smoker → cancer" is a specialization of "male → cancer", the set of database cases covered by the first rule are fully contained in the set of covered cases by the second rule, hence the 100%-indication to the right. The common set of database cases comprises 40% of the cases covered by the more general rule, hence the smaller indicator to the left.

Table 1
Example database from which two rules ("male → cancer" and "male ∧ smoker → cancer") were assessed and depicted in Fig. 9.

	male		female	
	smoker	no smoker	smoker	no smoker
cancer	60	15	75	10
no cancer	140	285	225	190

4.4. Temporal change. To present the temporal evolution of a rule set (w.r.t. the evaluation of selected measures), an animation is generated that displays the current state of the rule set at any given time (frame). Figure 6 depicts this idea with the same rule at three different times. If the consequent of the depicted rule is a failure class, this rule would be a candidate for a pattern that needs further investigation: the number of affected database cases (support) increased over time. The same can be stated for the lift and confidence. The latter means that the problem became more ane more severe since its probability increased.

However, the more data there is under analysis, the more patterns and thus, rules, can be found. It is not unusual to have several hundreds induced from a database. Clearly, this would clutter the visualization beyond recognition. We therefore proposed a method of thinning out the number of rules to be actually displayed [16]. This is done by allowing the user to linguistically specify what kind of temporal behavior he is interested in. For example, the user might be interested in rules showing a strong increase in support and moderate increase in confidence. Although the full description of this method is beyond the scope of this paper, the main idea is as follows: for every rule and every rule evaluation measure a time series is created. A trend analysis quantifies properties like increase, decrease or stability of these series (in terms of the slope of a regression line or other appropriate means). Fuzzifying the domains of these change rates and allowing the user to specify linguistic variables on these fuzzy partitions allow to calculate for every rule the degree of membership to the respective linguistic concept. A rule is then only depicted if its degree of membership exceeds a user-specified threshold.

4.5. Example scenario. Before we are going to apply the concepts introduced above on a real-world dataset, let us turn to an artificial example. We consider an automobile manufacturer that keeps record of the configuration of every car that leaves the production plant. Whenever a failure occurs, the database is updated. We assume for simplicity that the database contains only five attributes: air conditioning type (type 1 to 5), engine type (type 1 to 3), country (where the car was delivered to and where it was operating: Germany, Oman, Egypt, Norway, Iceland), time (of failure, discretized to 3 time frames named Jan, Feb and Mar), and failure (yes or no).

The intended pattern that shall be incorporated into this fictitious dataset is that in two countries (Oman and Egypt) one air conditioning type (type 1) is going to fail more and more often. The engine shall not have an influence on the failure rate. The general failure rate of all non-suspicious database entries was set to 15% with some random noise added, that is the failure rates lie between 83% and 87%.

The failure rates of the two cases (A/C 1, Oman) and (A/C 1, Egypt) grow from 17% via 30% to 40% and from 14% via 25% to 35%, respectively.

We visualize all possible rules with consequent failure=yes and antecedent referring to attributes air conditioning type and country. Since both antecedent attributes have 5 values each, we are going to visualize 25 rules.

Figure 7 depicts the rule set at all three times. The two outlier rules have been marked with a bold circle. Figure 11 shows the trajectory of both rules to give a better imagination of the motion. One can clearly observe the increase in the lift since the rule antecedent became more and more decisive for failure. There is, of course, motion among the remaining rules, however, the largest dislocation is obtained by the two outlier rules.

4.6. Application. We are now going to provide empirical evidence that the proposed visualization technique can support and simplify the data analysis process. As motivated in Subsec. 4.5, we now visualize data from a real-world car manufacturer. The dataset under analysis contains approximately 300 000 cases that exhibit 180 attributes. Since this data set was issued by a industrial partner, we are not allowed to provide confidential information such as the meaning of the attribute values. All we can tell is, that every tuple in the dataset represents a unique car that left the production plant of the manufacturer. Since for every car the time of a failure was logged as well, we were able to partition the full set of cars into data sets of (in this case) equal length. We used a preprocessing technique [16] based on Bayesian networks [5] to induce a set of attributes that should serve as antecedent at-

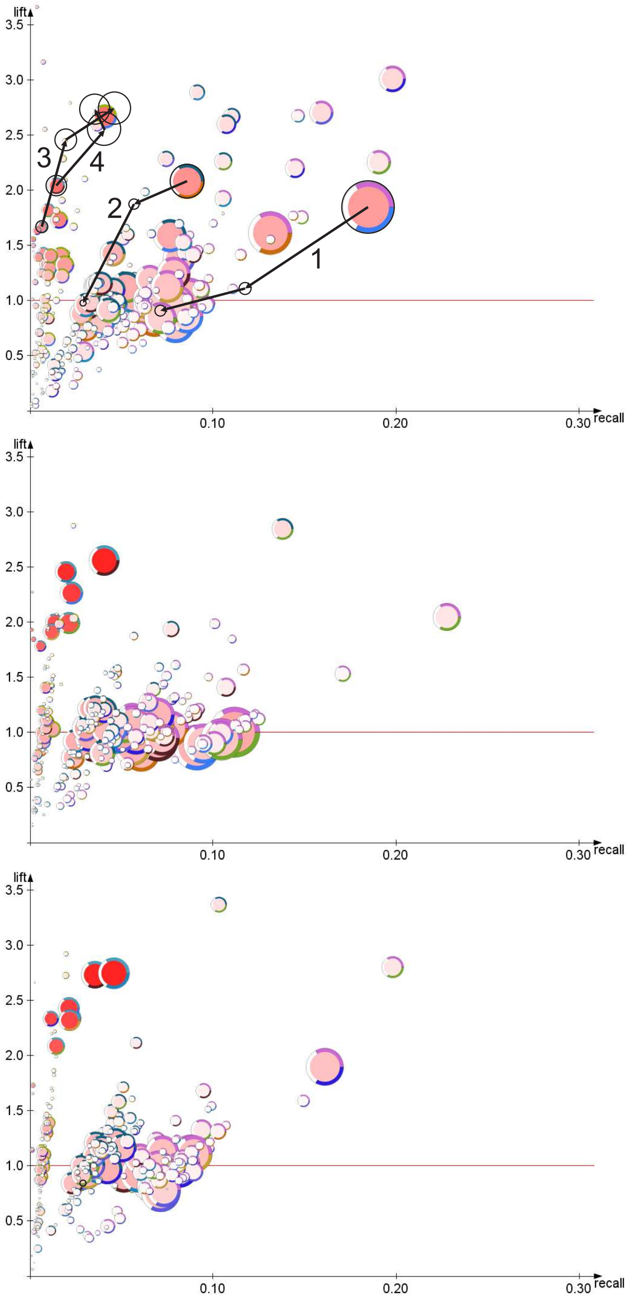

Fig. 10. Real-world application of a set of vehicles with a binary class variable: failure and no failure. Only rules indicating a failure are depicted. Two attributes were used to form the rules (hence two filled regions in the outer ring of every rule), therefore no overlapping of covered database cases could occur. The three charts shows the rules at the beginning, the middle and the end of the production period. To assess the motion of the rules, we superposed the final locations of the rules with the first image and indicated the corresponding rule with an arrow. Bold arrows indicate the four rules that were found interesting by experts.

tributes of the rules to be visualized. It was possible to identify a small set of meaningful attributes (out of the 180) that were used to generate rules whose temporal trajectories were visualized. Figure 10 shows a set of 760 rules at three different times. There are numerous rules that were interesting to experts. We selected four rules, two showing an evolving problem and two representing a vanishing group of failed cars. To simplify the assessment, we superposed the rule locations of the second and third chart with the first and indicated the motion with arrows. Four rules that showed an interesting behavior and could be assigned a meaning by experts are numbered in the figure: rule 1 and 2 represent shrinking sets of cars whose confidence is also dropping. More interesting, however, is the rapidly lessening lift which gave rise to the conjecture that the cause for the failure had been successfully addressed. Contrary, rules 3 and 4 represent sets of cars with increasing failure rate (confidence is increasing indicated by the darkening of the interior of the icons).

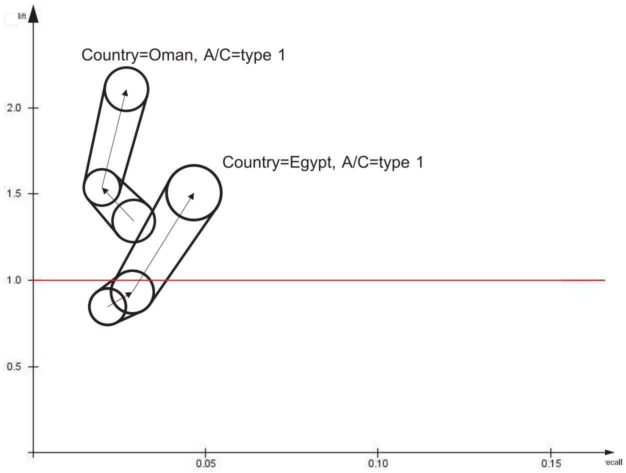

Fig. 11. Trajectory of the to outlier rules discussed in Subsec. 4.5. The individual rule sets at each time (Jan, Feb and Mar) can be seen in Fig. 7.

5. Conclusions

Visual data analysis tools provide a valuable tool to manage and process the overwhelmingly large information flow that a data mining task may return. In this article we gave an overview on two such approaches that visualized the local structure of graphical models and that postprocessed a set of association rules with the help of one or more linguistic concepts. Both approaches have been successfully applied to real-world problems in the automotive industry. The latter technique was designed to be independent of the origin of the rules that are to be analyzed. Hence, it is compatible with any rule-inducing algorithm and consequently reduces the effort when it is to be incorporated into an existing production system. A typical association rule induction algorithm finds so-called frequent item sets and uses these in a subsequent step to induce rules. Often, the frequent item sets are sufficient to users (e.g. if there is no dedicated class or failure variable). For such cases, the presented methods do not work directly. It is a current state of work to come up with modifications of the visualization methods that are applicable for general item sets.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A.N. Swami, "Mining association rules between sets of items in large databases", *Proc. ACM SIGMOD Int. Conf. on Management of Data*, Washington 1, 207–216 (1999).

[2] E. Castillo, J.M. Gutiérrez, and A.S. Hadi, *Expert Systems and Probabilistic Network Models*, Springer Verlag, 1997.

[3] J. Pearl, "Aspects of graphical models connected with causality", *49th Session Int. Statistics Institute* 1, CD-ROM (1993).

[4] S.L. Lauritzen and D.J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems", *J. Royal Statistical Society* B2 (50), 157–224 (1988).

[5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, 1988.

[6] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo, "Fast discovery of association rules", *Advances in Knowledge Discovery and Data Mining* 1, 307–328 (1996).

[7] Y.Y. Yao and N. Zhong, "An analysis of quantitative measures associated with rules", *Methodologies for Knowledge Discovery and Data Mining* 1, CD-ROM (1999).

[8] G.F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data", *Machine Learning* 9, 309–347 (1992).

[9] D. Heckerman, D. Geiger, and D.M. Chickering, "Learning Bayesian networks: the combination of knowledge and statistical data", *Microsoft Research, Advanced Technology Division* MSR-TR-94-09, CD-ROM (1995).

[10] C. Borgelt and R. Kruse, "Some experimental results on learning probabilistic and possibilistic networks with different evaluation measures", *1st Int. Joint Conf. on Qualitative and Quantitative Practical Reasoning* 1, 71–85 (1997).

[11] C. Borgelt and R. Kruse, "Probabilistic and possibilistic networks and how to learn them from data", *Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications* 1, 403–426 (1998).

[12] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation", *2000 ACM SIGMOD Intl. Conference on Management of Data* 5, 1–12 (2000).

[13] B. Goethals and J. van den Bussche, "On supporting interactive association rule mining", *Proc. Second Int. Conf. on Data Warehousing and Knowledge Discovery* 1874, 307–316 (2000).

[14] M.-C. Chen, "Ranking discovered rules from data mining with multiple criteria by data envelopment analysis", *Expert Systems with Applications* 33 (4), 1110–1116 (2007).

[15] M. Müller, "Entdeckung interessanter Assoziationsregeln", *Master's Thesis*, Otto-Friedrich-Universität Bamberg & DaimlerChrysler Research and Technology, RMI/DM, Ulm, 2005.

[16] M. Steinbrecher and R. Kruse, "Visualization of local dependencies of possibilistic network structures", *Granular Computing: at the Junction of Rough Sets and Fuzzy Sets* 224, 93–104 (2008).