Interactivity Closes the Gap

Lessons Learned in an Automotive Industry Application

Axel Blumenstock
Department of Applied Information Processing
University of Ulm, Germany
axel.blumenstock@uni-ulm.de

Jochen Hipp, Steffen Kempe,
Carsten Lanquillon, Rüdiger Wirth
DaimlerChrysler Group Research
Ulm, Germany
{jochen.hipp, steffen.kempe,
carsten.lanquillon,
ruediger.wirth}@dcx.com

ABSTRACT

After nearly two decades of data mining research there are many commercial mining tools available, and a wide range of algorithms can be found in the literature. One might think there is a solution to most of the problems practitioners face. In our application of descriptive induction on warranty data, however, we found a considerable gap between many standard solutions and our practical needs. Confronted with challenging data and requirements such as understandability and support of existing work flows, we tried many things that did not work, ending up in simple solutions that do. We feel that the problems we faced are not so uncommon, and would like to advocate that it's better to focus on simplicity—allowing domain experts to bring in their knowledge—rather than on complex algorithms. Interactivity and simplicity turned out to be key features to success.

1. INTRODUCTION

An air bellow bursts: This happens on one truck, on another it does not. Is this random coincidence, or the result of some systematic weakness?

Questions like these have ever been keeping experts busy at DaimlerChrysler's After Sales Services. Recently, they have attracted even more attention, when Chrysler's CEO LaSorda introduced the so-called tag process: a rigorous quality enhancement initiative that once more mirrors the enormous business relevance of fast problem resolution [3].

This primary goal of quality enhancement entails several tasks to be solved:

- detecting upcoming quality issues as early as possible
- explaining *why* some kind of quality issue occurs and feeding this information back into engineering
- isolating groups of vehicles that might suffer a certain defect in the future, so as to make service actions more targeted and effective.

Our research group picks up common data mining methods and adapts them to the practical needs of our engineers and domain experts. This contribution reports on the lessons learned. In particular, we elaborate on our experience that the right answer to domain complexity need not be algorithmic complexity—but rather simplicity. Simplicity opens ways to create an interactive setup which involves experts without overwhelming them. And if truly involved, an expert will understand the results and turn them into action.

We will outline the problem setting in Section 2. The subsequent sections respectively discuss the theoretical aspects, tool selection and model building methods, each answering the questions of what we tried and what finally worked.

2. DOMAIN AND REQUIREMENTS

2.1 The Data

Most of the data at hand is warranty data, providing information about diagnostics and repairs at the garage. Further data is about vehicle production, configuration and usage. All these sources are heterogeneous, and the data was not collected for the purpose of causal analyses. This raises questions about reliability, appropriateness of scale, and level of detail. Apart from these concerns, our data has some properties that make it hard to analyze, including

Imbalanced classes: The class of interest, made up of all instances for which a certain problem was reported, is very small compared to its contrast set. Often, the proportion is way below 1 %.

Multiple causes: Sometimes, a single kind of problem report can be traced back to different causes that produced the same phenomenon. Therefore, on the entire data set, even truly explanatory rules show only modest qualities in terms of statistical measures.

DMBA'06, August 20, 2006, Philadelphia, Pennsylvania, USA.

Semi-labeledness: The counterpart of the positives is not truly negative. If there is a warranty entry for some vehicle, it is (almost) sure that it indeed suffered the problem reported on. For any non-positive example, however, it is unclear whether it carries problematic properties and may fall defective in near future.

High-dimensional space of influence variables (1000s)

Influence variables interact strongly: Some quality issues do not occur until several influences coincide. And, if an influence exists in the data, many other non-causal variables follow, showing positive statistical dependence with the class as well.

True causes not in data: By chance, they are concludable from other, influenced variables.

2.2 The Domain Experts and Their Tasks

Our users are experts in the field of vehicle engineering, specialized on various subdomains such as engine or electrical equipment. They keep track of what goes on in the field, mainly by analyzing warranty data, and try to discover upcoming quality issues as early as possible. If they recognize a problem, they strive for finding out the root causes in order to address it most accurately.

They have been doing these investigations successfully over years. Now, data mining can help them to better meet the demands of fast reaction, well-founded insight and targeted service. But any analysis support must fit into the users' mindset, their language, and their work flow.

The structure of the problems to be analyzed varies substantially. This task requires inspection, exploration and understanding for every case anew. Ideally, the engineers should be enabled to apply various exploration and analysis methods from a rich repository. And it is important that they do it themselves, because no one else could decide quickly enough whether a certain clue is relevant and should be pursued, and ask the proper questions. Finding out reasons of strange phenomena requires both comprehensive and detailed background knowledge.

Yet, the engineers are not data mining experts. They could make use of data mining tools out of the box, but common data mining suites already require deeper understanding of the methods. Further, the users are reluctant to accept any system-generated hypothesis if the system cannot give exact details that justify this hypothesis. The bottom line is that penetrability and, again, interactivity are almost indispensable features of any mining system in our field.

3. UNDERSTANDING THE TASK

Let us first have a theoretical look at the problem. It is noteworthy that we will meet the following arguments again when we investigate individual methods.

3.1 What we tried

A great portion of the task can be seen as a classification problem. We would like to separate the good from the bad. It may be possible to tell for any vehicle whether it might encounter problems in the future. And if we choose a symbolic method, we can use the model to explain the problem.

As stated above, however, data is semi-labeled, and the problem behind the positive class may have multiple causes. These properties act as if there were a strong inherent noise that changes the class variable in either direction. Classifier induction tries to separate the classes in the best possible way but can return unpredictable, arbitrary results when noise increases. For our application, it suffices to grab the most explainable part of the positives and leave the rest for later investigation or, finally, ascribe it to randomness. In other words, we experienced that anything beyond *partial* description is not adequate here (confer Hand's categorization into mining the complete relation versus local patterns [5]).

So we came up with subgroup discovery (e.g., [8]). It means to identify subsets of the entire object set which show some unusual distribution with respect to a property of interest—in our case, the binary class variable.

Results from subgroup discovery approaches need not be restricted to knowledge acquisition, but can be re-used for picking out objects of interest. This is the partial classification we want, where a statement about the contrast set is not adequate or required.

Still, data properties make subgroup discovery results unusable most of the time. There are many candidate influences, and they interact strongly. Therefore, even if the cause could be described by a sole variable, it would be hard to find it among the set of variables influenced by it. All these variables, including the causal one, would refer to roughly the same subset of vehicles with an increased proportion of positives.

3.2 What works

Opposing it to mere discovery, we'd rather like to talk about subgroup *description*. It is to identify the very same subgroups, but in a way as comprehensive and informative as possible. The rationale is, even if subgroup *discovery* results are presented in a human-readable form, the users are left alone to map these results to synonyms that can be more meaningful in the context of the application. In a domain with thousands of influence variables, however, the users cannot be expected to bear all the (possibly even multivariate) interactions in their minds. Vehicle configuration, for instance, contains hundreds of strongly interrelated variables, dependent as well on type, production date and destination region. Subgroup description is thus required to provide any reasonable explanation as long as there is no evidence that the finding is void or unjustified.

4. A TOOL THAT SUITS THE EXPERTS

4.1 What we tried

We had a look at several commercially available data mining suites and tools. Unfortunately, any of these fell short of the requirements outlined in Section 2.2.

As an overall observation, they were rather inaccessible and often did not allow for interaction at the model building level. Even if they did, they could not present information (like measures) in the non-statistician users' language. Tools of this kind offer their methods in a very generic fashion so that the typical domain expert does not know where

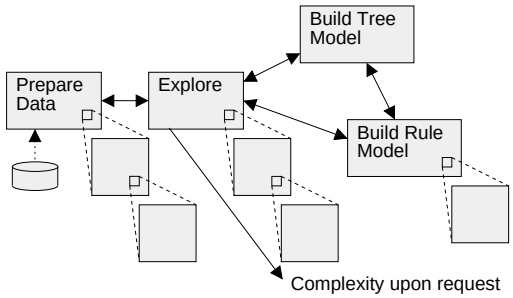

Figure 1: Coarse usage model of our tool. There is a fixed process skeleton corresponding to the original workflow. The user can just go through, or gain more flexibility (and complexity) upon request.

to start. In short, we believe that the goal conflict between flexibility and guidance can hardly be solved by any general-purpose application, where the greatest simplification potential, namely domain adaption, remains unexploited.

4.2 What works

We ended up in programming a tool of our own. Figure 1 shows a simplified view of our tool's process model. It emerged as the union of our experts' workflows und thus offers guidance even for users not overly literate in data mining. At the same time, it does not constrain the user to a single process but allows going deeper and gain flexibility wherever the user is able and willing to.

For example, the users start with extracting data for further analysis. We tried to keep this step simple and hide the complexities as much as possible. The user just selects the vehicle subset and the influence variables he likes to work with. A meta data based system cares about joins, aggregations, discretizations or other data transformation steps. This kind of preprocessing is domain specific, but still flexible enough to adapt to changes and extensions.

In the course of their analyses, the experts often want to derive variables of their own. That way, they can materialize concepts otherwise spread over several other conditions. This is an important point where they introduce case-specific background knowledge. The system allows them to do so, up to the full expressiveness of mathematical formulas.

A similar fashion of multi-level complexity is offered for the "Explore" box in Figure 1: The system offers both standard reports, suiting the experts' needs in most of the cases, up to individually configurable diagrams. For the sake of model induction, our tool offers currently two branches that interact and complement each other: decision trees and rule sets.

5. INTERACTIVE DECISION TREES

Subgroup discovery (and description) can be mapped to partitioning the instance set into multiple decision tree leaves. Paths to leaves with a high share of the positive class provide a description of an interesting subgroup. In fact, decision tree induction roughly corresponds to what our experts had been doing even before getting in touch with data mining. Hence, decision trees were the first method we chose.

5.1 What we tried

To quickly provide the users with explanation models, it was proximate to build decision trees automatically as is typically done when inducing tree-based classifiers ([2, 6, 11]). However, the experts deemed the results unusable most of the time, because the split attributes that had been selected by any of the common top-down tree induction algorithm were often uninformative or meaningless to them: The top-ranked variable was rarely the most relevant one.

For some time, we experimented with different measures. Literature suggests measures such as information gain, information gain ratio, χ^2 p-value, or gini index, to mention the most important ones.

However, in an exemplary analysis case, there was a variable that gave the actual hint for the expert to discover the quality issue's cause. This variable was ranked 27th by information gain, 41st by gain ratio, 36th by p-value and 33rd by gini index. We conclude that an automatic induction process hardly could have found a helpful tree.

5.2 What works

This is where interactivity comes into action. This is close what Ankerst proposed [1], except for the mining goal. Building trees interactively relieves the measure of choice from the burden of selecting the single "best" split attribute. The idea is almost trivial: Present the attributes in an ordered list and let the expert make tentative choices until he finds one he considers plausible.

What remains is the problem of how to rank the attributes in a reasonable way. But even for ranking, the aforementioned statistical measures proved little helpful. We explain this by the fact that they are measures designed for classificator induction, trying to separate the classes in the best possible way. But as illustrated in Section 3, this is not the primary goal in our application.

Most of the time, we deal with two-class problems anyway: the positive class versus the contrasting rest. Hence, we can use the measure *lift* (the factor by which the positive class rate in a given node is higher than the positive class rate in the root node). To complement the lift value of a tree node, we use the *recall* (the fraction covered) of the positive class. Both lift and recall are readily understandable for the users as they have immediate analogies in their domain. Now, focusing on high-lift paths, the users can successively split tree nodes to reach a lift as high as possible while maintaining nodes with substantial recall.

In order to condense this into a suitable attribute ranking, we group attribute values (or, the current node's children). We require the resulting split to create at most k children, where typically $k = 2$ so as to force binary splits. This ensures both that the split is "handy" and easily understood by the user, and that the subsequent attribute ranking can be based consistently on the child node with the highest lift.

To group the children in a reasonable way, we simply sort

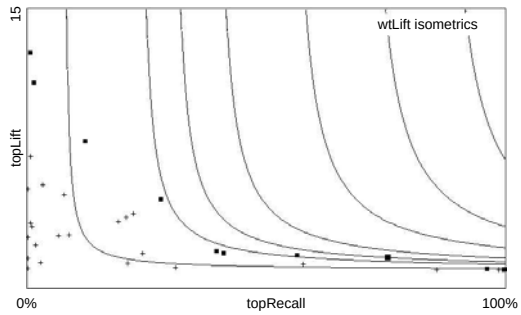

Figure 2: Quality space for the assessment of split attributes. Each dot represents an attribute, plotted over recall (x axis) and lift (y axis) of the best (possibly clustered) child that would result. Dots are plotted bold if there is no other dot that is better in both dimensions. The curves are isometrics according to the recall-weighted lift.

them by lift. Then, keeping their linear order, we cluster them using several heuristics: merge smallest nodes first, merge adjacent nodes with lowest lift difference. Lift and recall of the resulting highest-lift node are finally combined to a one-dimensional measure (weighted lift, or "explanational power") in order to create the ranking.

Grouping is automatically performed during attribute assessment. Still, the users can interactively undo and redo the grouping or even arrange the attribute values into any form that they desire. This is important to further incorporate background knowledge, e.g. with respect to ordered domains, geographical regions, or, in particular, components that are used in certain subsets of vehicles and should, thus, be considered together.

As an alternative to a ranked list, the user can still get the more natural two-dimensional presentation of the split attributes (Figure 2). Similar to within a ROC space, every such attribute is plotted as a point. We use recall and lift as the two dimensions.

6. INTERACTIVE RULE SETS

As an important data property we mentioned that influences interact in a way that some quality issues do not occur until several influences coincide. While decision tree building is intuitive, its search is greedy and thus may miss interesting combinations. So the experts asked for an automatic, more comprehensive search. This led us to rule sets.

6.1 What we tried

Among others (e.g., [7, 13, 12]), a well-known subgroup discovery algorithm is CN2-SD [9]. It induces rules by sequential covering: By heuristic search, find a rule that is best according to some statistical measure. Reduce the weights of the covered examples, and re-iterate until no reasonable rule can be found any more.

The first handicap of this procedure is the same as with decision trees: There is no measure that could guarantee to select the best influence, here: rule.

But even the hope that a good rule will be among the subsequently mined ones need not hold: Imagine there are two rules describing exactly the same example set. CN2-SD will never find both, because by modifying the examples' weights, the two rules' ranks will change simultaneously. This, however, runs counter the idea of subgroup *description*, in other words, comprehensiveness at the textual level rather than mere subset identification.

6.2 What works

We thus came up with an exhaustive search (within constraints). It is realized by an association rule miner with fixed consequence. This is not new, and like us, many research groups think about how to handle redundancy within the results (e.g., [4, 10, 14])

What we like to point out here is that once again, the idea of interactivity produced a simple but effective solution. The expert is enabled to control a CN2-SD like sequential covering. He picks a rule he recognizes as "interesting" or "already known". This is comparable to selecting a decision tree split attribute. Several measures, fitting into his mindset, support him with his choice. The instance set is then modified so as to remove the marked influence, and the expert can re-iterate to find the next interesting rule.

7. MODULE INTERACTION

The key property that makes a tool more than the sum of its components, however, is the facility of interaction between its exploration and modelling components. This is still only partly implemented, but our users strongly request for it. Indeed module interaction is the feature that allows them to flexibly apply the methods offered and to take out the respective best of them.

Such sometimes trivial but practically important features include:

- Extracting instance subsets as covered by a rule or tree path and exchanging them within the modules for deeper analyses or visualization.
- Building a tree with a path as described by a rule in order to take a closer look at the respective contrast sets.
- Deriving new variables from tree paths or rules.

8. CONCLUSION

We reported on our experiences of applying data mining methods in a domain where data is difficult, analysis tasks change structurally case by case, and thus a great amount of background knowledge is indispensable. Many approaches suggested in the literature turned out either too constrained or too complex to be offered without major adaption. In such a setting, we consider it best to stick to simple methods, provide these in a both flexible and understandable way, and settle on interactivity.

Still, there is a wide field to explore. At many points of the process, there is much room for methods that support the experts and reduce their routine work load as much as possible.

9. REFERENCES

[1] M. Ankerst, C. Elsen, M. Ester, and H.-P. Kriegel. Visual classification: an interactive approach to decision tree construction. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall, 1984.

[3] M. Connelly. Chrysler's LaSorda on quality: Fix it now. *Automotive News*, May 9th 2005.

[4] F. Gebhardt. Choosing among competing generalizations. *Knowledge Acquisition*, (3), 1991.

[5] D. J. Hand. Data mining—reaching beyond statistics. *Research in Official Statistics*, (2):5–17, 1998.

[6] G. V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29:119–127, 1980.

[7] W. Klösgen. EXPLORA: A multipattern and multistrategy discovery assistant. In *Advances in Knowledge Discovery and Data Mining*, pages 249–271. 1996.

[8] W. Klösgen. Applications and research problems of subgroup mining. In *Proceedings of the 11th International Symposium on Foundations of Intelligent Systems*, 1999.

[9] N. Lavrač, P. A. Flach, B. Kavšek, and L. Todorovski. Rule induction for subgroup discovery with CN2-SD. In *ECML/PKDD'02 Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning*, 2002.

[10] B. Liu, M. Hu, and W. Hsu. Multi-level organization and summarization of the discovered rules. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.

[11] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, 1993.

[12] G. I. Webb. OPUS: An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research*, 3, 1995.

[13] S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery*, 1997.

[14] X. Yan, H. Cheng, J. Han, and D. Xin. Summarizing itemset patterns: a profile-based approach. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2005.