When Distribution is Part of the Semantics: A New Problem Class for Distributed Knowledge Discovery

Rüdiger Wirth, Michael Borth, Jochen Hipp

DaimlerChrysler AG,
Research & Technology
PO BOX 2360, 89013 Ulm,
Germany
{ruediger.wirth, michael.borth}
@DaimlerChrysler.com

Abstract. Within a research project at DaimlerChrysler we use vehicles as mobile data sources for distributed knowledge discovery. We realized that current approaches are not suitable for our purposes. They aim to infer a global model and try to approximate the results one would get from a single joined data source. Thus, they treat distribution as a technical issue only and ignore that the distribution itself may have a meaning and that models depend on the context in which they were derived. The main contribution of this paper is the identification of a practically relevant new problem class for distributed knowledge discovery which addresses the semantics of distribution. We show that this problem class is the proper framework for many important applications in which it should become an integral part of the knowledge discovery process, affecting the results as well as the process itself. We outline a novel solution, called Knowledge Discovery from Models, which uses models as primary input and combines content driven and context driven analyses. Finally, we discuss challenging research questions, which are raised by the new problem class.

Introduction

At DaimlerChrysler Research and Technology, we recently started a research project, called CarMining, which focuses on distributed knowledge discovery (DKD). For the purpose of this paper, its most important aspect is that we view vehicles as mobile data sources and we want to mine these distributed data sources.

Working on this project, we realized that all the approaches described in the literature are not really suited for our problem. Although one could map our problem to distributed knowledge discovery approaches, there remains an important aspect missing, which is crucial in our application scenario. This missing aspect is the assumption of all known approaches that all data sources have equal semantics. They aim at inferring a global model and try to approximate the results one would get, if one could join all the data sources. Distribution is considered merely a technical issue. It is ignored that the distribution itself has a meaning and that the components of a model are dependent on the context in which they were derived. We will argue, that that ignoring the meaning of distribution is not appropriate in many practically important application domains and that it will lead to wrong results.

The rest of the paper is organized as follows. First, we will describe our project, its practical needs and the application problem we want to solve. Then, we briefly review the literature on DKD and provide arguments why current DKD approaches are not suitable for this application. From this, we derive a new problem class which we call *Semantically Distributed Knowledge Discovery*. We claim, that this problem class is not restricted to our project but that it is also the proper framework for many other important application areas. Finally, we outline a novel solution for this problem class that we will pursue in our project.

The main contribution of this paper is the identification of a practically relevant new problem class for distributed knowledge discovery, which has not yet been addressed by the research community. We phrase some research questions and point to open issues which must be addressed.

Vehicles as Distributed Data Sources

The project CarMining originated in problems identified during earlier data mining projects on the analysis of quality information about Mercedes-Benz vehicles. One of the main tools for monitoring and assessing the quality of the vehicles is the database QUIS (QUality Information System) that stores information on all warranty cases of all Mercedes vehicles for several years. Quality engineers use QUIS to identify potential quality problems which are

then quickly addressed and adequately solved. For solving the problems, the engineers typically use many other additional information sources such as reports from repair shops, design documentation, and trial reports. With data mining, we support the engineers in identifying problems quicker and in constraining the set of affected cars.

While successful in some applications, we encountered several problems. First, data arrive fairly late at QUIS, because most problems are either discovered at routine inspections or when a fault is recognized by the customer. Thus, it takes time until quality problems surface, i.e., there is a delay leading to unnecessary production of potentially flawed vehicles. Second, the technical information on warranty cases in QUIS is not detailed enough to pin down the reason for the fault precisely, This means that the engineers need more time to explore more potential causes and perhaps introduce costly improvements in more vehicles than needed. The third problem concerns the quality and reliability of the data. The repair shops enter most of the data manually which leads to many data errors.

A solution for these issues would be to collect quality related information in the vehicle directly and transmit them when needed. Since the amount of potentially relevant data in a vehicle is immense, we need to perform a large part of the data mining already in the vehicle in order to compress the data to get higher order information. Thus, we need to extract relevant knowledge from vehicles, i.e., distributed data sources.

Questions we want to answer include the following:

- What are the dependencies between special equipment and faults?
- Do certain combinations of special equipment lead to significantly higher faults of some components?
- Under what conditions does a part/component break down?
- Under what conditions is a driver assistance system (e.g., ESP, ABS)[1] activated?

In the past, we addressed these questions with different settings such as classification and clustering (Wirth and Reinartz, 1996). So far, the best results were achieved by formulating this as a dependency modeling problem, using association rules (Hipp and Lindner, 1999) and Bayesian networks learned by a K2 like algorithm (Cooper and Herskovits, 1993; Borgelt et al., 1998).

This project raises many challenging applied research questions such as communication, data processing in the vehicle, and scalability to millions of vehicles. Furthermore, we are very concerned with the privacy of the data collected and the security of the transmission. However, for the purpose of this paper, we focus on one aspect, the semantics of the distribution of the data sources.

Limitations of Current Approaches to Distributed Knowledge Discovery

Distributed Knowledge Discovery (DKD) is a very active sub area of data mining with contributions from several different scientific fields. Kargupta and Chan (2000) edited a collection of research papers which are representative of the state of the art. Unfortunately, none of the currently known approaches can solve our problem adequately. In this section, we briefly review the literature and then explain, why these approaches are deficient for our purpose.

Current Approaches to Distributed Knowledge Discovery

DKD embraces the growing trend of merging computation with communication. Telematics and ubiquitous computing become increasingly important. DKD accepts the fact that data may be inherently distributed among different sites. It offers techniques to discover new knowledge through distributed data analysis and modeling using minimal communication.

Until recently, research on distributed knowledge discovery has been motivated primarily by the desire to scale up to very large data bases (Provost, 2000). Since scaling up is not the primary issue in our application, we will neglect this aspect, as well as communication and security, in the discussion in this paper.[2]

By definition, DKD must deal with distributed data and there are different possibilities for the data distribution. Kargupta et al. (2000) distinguish between a *homogeneous* and a *heterogeneous* case. In the *homogeneous* case, tables at different sites have columns corresponding to the same set of features. In other words, there is a consistent database schema across all the sites. In the heterogeneous case, tables at different sites have columns corresponding to different features. In other words, there is no consistent schema across the different sites. Kargupta and his

[1] The ESP is the electronic stabilization system that intervenes in critical driving situations. ABS prevents the breaks from blocking.

[2] Of course, we need to address these issues in the implementation of our system, but they are not the main issue from the application point of view. The main issue is that our data sources are the vehicles which are inherently distributed.

colleagues developed methods, called collective data mining, to learn directly from heterogeneous data sources. The goal is to achieve the same result that one would get if the data could be combined.

Another prominent approach to DKD is meta learning (Chan and Stolfo, 1993; Stolfo et al., 1997) which attempts to derive a global classifier based on classifiers derived from local sources. The whole approach is geared towards the homogeneous case. If the data schemata are incompatible, Prodromidis et al. (2000) propose methods to overcome this problem by bridging the gap between the two schemata to make them compatible.

Most work on DKD focuses on classification and segmentation tasks. But there are a few papers which address dependency analysis. Yamanishi (1997) and Lam and Segre (1997) describe distributed Bayesian learning algorithms for homogeneous data sets. Both approaches attempt to estimate the global distribution and are geared towards scalability.

In summary, there are many different approaches to DKD, but all of them attempt to build a global model which approximates the model one would achieve if the data were combined. Essentially, they treat the distributed data sets as a single virtual table. As we will discuss in the next section, this basic assumption is not valid in our application, and, as we will show later, is not even true in many applications discussed in the literature.

Why Current Approaches are not suitable

Provost (2000) mentions that the current view of DKD may be too narrow and that DKD could be more than scaling up. He notes that databases *are* distributed and that there may be good reasons against a monolithic data base. And this is the case in our application.

At first sight, the CarMining project looks like a straightforward application of DKD. Each of the vehicles shares the same set of features, so the methods from the homogeneous case of DKD seem to be appropriate. As we will show this straightforward approach is not appropriate in our scenario, where the different sites under investigation are the vehicles. A closer look reveals that the distribution of the data over different sites is a fundamental part of the semantics behind our mining scenario. Therefore, it needs a special treatment which none of the current approaches offers.

The underlying basic assumption of all DKD approaches is that there is a global model, which could be computed if all data were combined. They try to approximate this universally best model through the combination of base models derived from distributed data sources. If a feature is present in different data sources, it is assumed that this feature has the same meaning everywhere. If not, pre-processing steps are required to assure this (Prodromides et al., 2000). If a feature-value combination appears in several data sources, this combination is considered to be identical, and each one contributes the same information for the final model. In other words, the distribution of the data sources is treated only as a technical aspect. It is ignored, that the distribution of the data itself may have a meaning.

In our application scenario, the meaning of the data depends heavily on their context. An observation in a vehicle must be interpreted in its proper context. For instance, a speed of 100 mph may be considered very high for a van, but moderate for some limousines. Furthermore, observations are not independent of each other. For instance, causal and temporal dependencies cause certain observations always to occur together.

Let us explore this further with an example from dependency analysis with association rules (Agrawal et al., 1993). Association rules model dependencies in the form of implications like $\{x_1,...,x_n\} \rightarrow y$. The meaning of such an expression is: "Whenever we observe $x_1,...,x_n$ then it is likely that we will also observe y". For instance, we might find the rule "$Speed > 100mph \wedge SteeringAngle > 20^o \rightarrow ESP$". The rule may hold with confidence 90%. This means, in nine of ten cases when speed and steering angle are above 100 mph and 20 degrees, respectively, the ESP starts supporting the driver.

It is obvious that the existence of such a dependency highly depends on additional circumstances. For instance, this rule might hold for a van but not for a limousine. The latter may show this behavior only at much higher speeds and higher steering angles. Taking into consideration that limousines outnumber vans on the streets, it is evident that the rule above, holding mainly for the minority vehicle model, will not occur with an adequately high confidence in the results set. Thus, we would miss an important dependency.

One solution for this simple case could be to add an attribute to each data record that describes the vehicle model. Unfortunately, generalizing this solution is not feasible. There is a huge number of possible attribute combination that could make up the context. This will become even worse if we consider external information like weather conditions or regional characteristics which might influence the vehicles behavior. We cannot specify all attributes in advance that describe all possible contexts that might be important.. And even if we could, adding that many attributes would make the search space explode and would clutter the results sets with coincidental patterns.

As mentioned above, conventional DKD treats all the distributed data sets as a single virtual table. Context information is ignored if not explicitly added to each record.

Phrasing the problem in terms of classification does not help either. Consider the meta learning approach (Chan and Stolfo, 1993), which looks attractive at first sight. The vehicles could be viewed as homogeneous data sources in that the same features are monitored in each vehicle. However, as discussed above, the meaning of the values of each feature depends on the context. The local classifier could capture this context. The problem arises, when the output of this local classifier is used in meta-learning. In essence, it would vote in situations where it is not applicable. The averaging out effect of this ensemble learning would not be sufficient. The classifiers differ not randomly which is an important requirement for the success of ensemble learning (Dietterich, 2000), but systematically.

The collective data mining approach (Kargupta et al., 2000) for heterogeneous data sets is closest to the requirements of our situation, but suffers from the same fundamental problem. The base classifiers are combined without taking the semantics of the distribution of the features and the observations into account.

A New Problem Class for Distributed Knowledge Discovery

Problem Characterization

In this section, we describe the new problem class, *Semantically Distributed Knowledge Discovery*. We don't want to elaborate on a formal definition of the semantics of the distribution, because this depends on both the representation formalism and the application problem. This would not really help at this stage. Instead, we provide an intuitive characterization.

There are two situations, when distribution is an integral part of the semantics of the problem. First, the meaning of at least some of the features depends on the context, i.e., some features mean different things at different data sites. Secondly, instances (or observations) are present at a site for a particular reason. This could be because the site attracted the instance for some reason or that the instance only makes sense in combination with other instances at a site.

Alternatively, we can view this the other way round. If it doesn't matter, whether a data record occurs at one site or an other, then the distribution is meaningless. Otherwise, the distribution has a meaning and needs to be explicitly considered in a solution.

Let us assume, we have n data sites $D_1,...,D_n$. At each data site D_i, there are m_i features $F_{i1},.....F_{im_i}$ which need not be the same (or have the same meaning or granularity) at all sites. $I_{ik},.....I_{ik_i}$ denote the instances at data site D_i, which are described as vectors of the features $F_{i1},.....F_{im_i}$.

Let us also assume, there is some target concept C_g to be learned. For the purpose of our discussion, a concept can be anything, including a classifier, a segmentation model, a dependency graph, and a probability distribution. The goal of DKD is approximate C_g by $\hat{C}_g$ using some combination of local concepts C_i from each data site D_i.

In DKD, the target concept C_g is always the concept one would get if all data were combined at one site. In other words, DKD treats the distributed data sets as a single virtual table. If some data are transferred from one site to an other, then the learned concept $\hat{C}_g$ may be different, but the target concept remains the same, regardless of the distribution of the data.

We claim, that in many applications this assumption is not true. Often, the (true) target concept C_g would change if a few records were transferred from one site to an other. Then, the physical distribution corresponds to a semantic distribution, i.e., the location of the data has a meaning and needs to be explicitly addressed.

Models depend on the context in which they were derived. Thus, components of a model, e.g., single rules, branches of decision trees, or fragments of Bayesian networks, cannot be exchanged without taking this context into account.

If the location has a meaning, then it may be important to reason not only about records at a site but also about the sites themselves. For instance, we might need to know *why* certain things happen at one site and not at an other, or we might be interested in identifying subgroups of data sites where similar concepts hold.

Actually, such questions are the driving forces behind our CarMining application. We want to understand why certain parts break down in one vehicle while it performs well in the vast majority of cases. And we want to identify those vehicles where the same part will probably also break down in order to take preventive actions.

These are questions which cannot be addressed in a standard problem setting which uses a single huge virtual table. In our case, data are distributed for a reason and cannot be joined without semantic confusion. Furthermore, we need to combine information from different levels, the level of observations at a data site and the level of data sites themselves.

Please note that this differs from standard DKD scenarios where we have different data bases or tables for different aspects of a problem, e.g., customer data and transaction data that are stored in different parts of a company.

Application Scenarios

The problems discussed above are certainly important in our CarMining application. But we are convinced that this is a rather typical mining situation for distributed knowledge discovery and that the problem class above is the proper framework for many applications.

Let us consider a few typical applications which are often cited in DKD. Banking and fraud detection are popular examples. It is argued that banks would like to share their models for fraud detection or loan approval, Obviously, they cannot exchange or combine all data from their customers for privacy reasons. DKD is a compelling idea in this scenario. Build local models at each bank, perform some meta learning with these local models, and get a global classifier. There is no need to exchange raw customer data.

However, this straightforward approach ignores the fact that each bank is different. For instance, the rules for issuing a credit card may be different, the definition of fraud (or how much fraud is tolerated) may be different, and the demographics of customers may be different. A particular customer may have a loan at a bank since his credentials are in line with the policy of this particular bank while other banks rejected his loan application. This means, the local models are biased by many facets specific to each bank. Thus, combining the models would lead to the undesired situation that the policy of one bank affects the classification of customers at another bank, while this other bank may have completely different policies.

The health industry is another promising application domain for DKD. Hospitals or health insurance companies also can not exchange data on their patients for privacy reasons. Nevertheless, they (and other medical practitioners and researcher) would like to learn from each others experience. Again, DKD looks like a good solution. We can build local models at each hospital, exchange the models, and combine them to get a global model which is valid for all hospitals and which draws on the experience of all hospitals. However, combining the models in the typical DKD way will not produce valid results. Each hospital has its special areas of expertise. They differ in skills and equipment, and treat patients differently. Thus, they may attract different kinds of patients. A patient who survived a heart attack in a hospital specialized in cardiac diseases might have died at another hospital. Thus, combining the models would not work, because the models would not take this fundamental difference into account. The problem is that the distribution of the data has a meaning, and thus, this scenario is an instance of our problem class.

Conventional DKD would try to answer the question, under what conditions a patient survived a heart attack, focussing on features of the patient. In our problem setting, we are first interested, why a patient survived a heart attack at a specific hospital, and then we would try to find out why a patient with similar characteristics died at an other hospital. For the second step, characteristics of the hospital, i.e., the data site, are important. Thus, we want to learn something on both the patient *and* the hospital, and we need to have data on both the patient and the hospital.

A further example for our problem class is the supermarket domain. Consider the use of association rules to discover which items of a supermarket are bought together. Let us assume that a supermarket chain wants to analyze shopping patterns of many supermarkets. One approach would be to transfer all data to a central site, combine them, and compute the association rules from the combined huge data set. What we get there, would be shopping patterns common to most of the supermarkets based on the implicit assumption that all supermarkets are equivalent. However, this is certainly not the case. The supermarkets typically show fundamental differences based on issues like demographics, traffic situation, and local competition. There are also regional differences leading to different shopping behavior and different assortment of goods.

Common approaches to DKD (Zaki et al., 1996; Cheung et al., 1996) would avoid the need to combine all the data at one site or speed up the process through parallel computation, but they would miss the fundamental semantic difference between the supermarkets. Again, here we are interested in the shopping patterns in the proper context.

Open Issues

Semantically distributed knowledge discovery raises many interesting and challenging research questions. In general, we need methods, which take the distribution into account and make explicit use of it.

Theoretically, one could address this problem by explicitly representing the context. However, this is not a practically feasible solution. We cannot foresee all possible situations that might be important to capture the true target concept. Furthermore, adding more features to the local learning process would lead to combinatorial explosion and increase the number of coincidental patterns.

Instead, a different perspective is needed. The local models are important in their own right and need to get more attention. Local models implicitly represent the context and this needs to be taken into account when the models are combined. A voting scheme is certainly not sufficient. The process needs to be more elaborate. In section 5, we outline a solution approach we pursue in our project. But this is certainly not the only one and there remain many open research questions.

First of all, we need a deeper understanding of the problem class itself. This paper is only a preliminary investigation motivated by our application problem. A formal analysis of the semantics of distribution might provide more insights and suggestions for solutions. We need more detailed studies into the discovery, representation, and characteristics of semantic context in distributed knowledge discovery. Formalisms are needed that capture and measure context sensitivity to decide which context is relevant to a specific bit of information and how strong such relations are. Solutions to these questions may extend existing analyses that measure dependencies between local variables to groups of variables, more general forms of influences and comparisons between different sites. Interesting mining results may often hold only for subgroups of data sites and subgroups of observations at distributed data sites of the whole population or data sites. Work on identifying interesting subgroups (Kloesgen, 1996; Wrobel, 1998) is a starting point to address this problem, but needs to be adapted and extended.

Knowledge discovery is already an ill-defined process. Taking the semantics of the distribution into account makes it even worse. We now talk about different levels of analysis, i.e., analysis at the observation or object level and analyses at the level of data sites. Meta-learning and meta-analysis will get an extended meaning. The feature selection process will become much more complicated because different meanings and the context need to be taken into account. For all that, meta-data would be very useful throughout the process.

Models in a symbolic representation can explicitly represent local contexts. In our solution approach, we use Bayesian networks as local models. But what about other popular models like decision trees or decision rules? Decision trees are highly unstable, i.e., small changes in the observation can lead to drastic differences in the resulting tree. Can we use such unstable models in this distributed setting?

These are some of the issues that come up when we address the problem directly, and there are different ways we can approach this. In the following section, we present a solution that we pursue in the CarMining project.

Knowledge Discovery from Models

The requirement to address the semantics of distribution within the CarMining project led us to a new approach for DKD, which we call Knowledge Discovery from Models (KDM). In its current form, it is applicable whenever the local sites provide models based on symbolic knowledge representations, e.g., Bayesian networks.

Discovery from Local Models

We propose an abstract solution for this problem class that consists of the following four steps:

1. Build local models
2. Compare local models at a central site to identify interesting differences and similarities
3. Explain differences and similarities through additional analysis steps, for instance, to identify interesting subgroups.
4. Act on the insights, for instance, build executable models

We want to illustrate it briefly with the association rule example presented earlier. As mentioned above, adding attributes to the observation in order to describe the vehicles is not a practically feasible solution. But do we really need to describe the details of the vehicles in order to capture distributed semantics? To answer this question, we want to draw the attention to the actual goal that motivated us to try incorporating vehicle descriptions into the data sets. In fact, the reason was to enable the mining methods to distinguish subgroups or cluster of the vehicles. In other words, we aimed at making the mining methods aware of the vehicles. We can reach this goal indirectly by enabling the methods to distinguish between the different vehicles. The resulting subset building is not based on the characteristics of the attributes. It is based on enumeration of vehicles. Instead of learning that a rule holds only for the van-models of the population, an enhanced mining method should return the rule together with a set of vehicles

for which the rule holds. Deducing that this subset is (nearly) identical with the subset of the van-model vehicles is then up to further explanation or mining steps. For that purpose, we employ additional information available in the central mining station, e.g., coming from the QUIS database. A possibility is to set up a table describing the vehicles and to add a binary attribute indicating whether a vehicle belongs to an identified cluster or not. Then a decision tree algorithm can be employed to characterize the identified subset. It is important to note that the attributes further describing the vehicles are added after the actual rule generation phase. This allows exploration without blowing up the search space during rule generation.

This example also illustrates that we get the problem of scalability, although at a different level. The problem is not so much the number of observations at each site. A more serious problem arises when we have thousands of data sites. We expect a fundamental difference when we move from a few data sites, say up to a few dozens, to several thousands or even millions.

We do not claim that this is the only way to approach semantically distributed knowledge discovery and each of the four steps above already raises many challenging research questions.

As we already pointed out earlier, symbolic models constructed by local learning represent information within local context. Obviously, it depends on the goal of the analysis task which parts of a model represent the information relevant to the task and which represent the context relevant to the information. Symbolic models are a sensible input for knowledge discovery algorithms that address the semantic aspects of distributions, since such algorithms should derive knowledge about different pieces of information, about contexts in which these pieces of information hold true and about the subgroups of the input in which these contexts are given. One can address all these points by an identification and analysis of differences and similarities between the local models.

It is important to note that information, context and subgroup detection depend on each other. We can devise an algorithm that looks for subgroups of the data sites, i.e., vehicles in our case, that show a maximum correspondence of parts of their models. These corresponding parts represent the common local context of the respective subgroup**.** The algorithm then determines the information that is characteristic for these groups and relevant to the analysis task. Alternatively, we can work the other way round. The algorithm searches for the proper local context in which a given bit of knowledge occurs, e.g., by comparing all models that contain this information. Typically, we need a combination of both procedures. The following section provides an example.

KDM in CarMining

Within the CarMining scenario, each vehicle uses Bayesian networks for various local tasks such as onboard diagnosis. These networks are at least partially constructed by local learning and represent highly compressed information about the working condition and configuration of the vehicles. The vehicles transmit the networks to a central site, where they are stored in a data base. To solve the quality related analysis tasks described in Section 2, we select the appropriate networks from this data base and use them as input for our algorithms.

Our first software prototype aims at dependencies discovered on car data together with the information under which circumstances they occur and the respective groups of vehicles. Within Bayesian networks, dependencies are represented by edges between variables. Possible representations for the context are fragments of a network or probability distributions over states of variables. We prefer network fragments, since they express both dependencies and probabilities of states. This allows us to work on a single input type for information and context.

Let us assume, our task is to find details about the failure of a component, which is used in a variety of cars. From engineering we know that the component's temperature is an important factor, so we will disregard results without usable information about component failure or temperature. We construct fragments together with the groups of vehicles in which they occur by using a bottom-up approach which starts from minimal fragments. In a way similar to the Apriori algorithm (Agrawal and Srikant, 1994), we use intersections between the groups to determine which fragments occur together in a statistically significant number of vehicles. Eventually, the algorithm produces a set of results. Each consists of one or more network fragments representing failure dependencies as well as local context and a list of the vehicles in which these fragments occurred.

Examples for results could be the following:

- A failure is likely to occur for very high temperatures, independently of any other information,
- Components in vehicles of a certain configuration get very hot. As expected, this leads to failures. This result could be derived from a single connected network fragment. It is more specific than the first result and holds true for a proper subset of the vehicles of the first.
- The exact temperature for which the probability of failure increases significantly depends on the configuration. A statement like this can be derived from several results, each containing a fragment representing a specific configuration and a fragment for the dependency between temperature and failure.

The probability distributions of these dependencies differ. The meaning of “too hot for the component” depends on the context.

- For one configuration, there are two subgroups with different critical temperatures. This statement can be reflected by two results with the same fragment for the configuration. A proper explanation requires external data. Here, a closer look at production data may reveal a quality problem in a specific production line.

We store these results in a data base, including meta-information and details about their generation. We can use this data base later for comparisons between results, like trend analyses, and for improving our learning algorithms (e.g., by meta-learning).

The generated set of results offers insights on vehicles and on the appropriate segmentation of all the vehicles in groups where the context relevant to the quality problem is homogeneous. This fulfills the requirements of our project and we can provide engineering with the requested information. Additionally, since we now know a sensible segmentation, we can easily introduce a feedback cycle to improve local models or local learning. For this, we adjust local parameters according to information found for the respective segment of cars.

Work on this approach is in progress and there are several open research questions. Our approach, for example, generates a multitude of results that make it difficult for the user to identify the most relevant ones. We can assist the user with focusing techniques that use personalized preferences, but this does not exploit the existing similarities and differences between the semantics represented in the results. One approach we can envision today is to compute a semantic graph that shows such relations within the result multitude and use the focusing techniques to provide an entry point for a semantic navigation of the results. In general, we need better ways to *explain* the similarities and differences and to translate them in reasonable actions.

Summary and Conclusions

We showed that the semantics of distribution leads to a new problem class within distributed knowledge discovery for which current approaches produce insufficient results. It is necessary to find novel solutions that account for the major aspects of this new problem class: context sensitivity of data and generated models and the resulting non-interchangeability between different sites.

In this paper, we discussed a list of open questions which must be addressed by future research. In summary, we need to find sensible ways to work with the semantic aspects of distribution and solve the various problems along a very complex process. Since the new problem class is the proper framework for various application areas, we expect advances in several directions, leading to both improved insights in the application domains and more experience how to deal with this problem class.

We presented our own approach, knowledge discovery from models, that was motivated by both research questions and project considerations: It maintains the context of information within the knowledge discovery process and makes effective use of local models, in our case Bayesian networks in vehicles.

The many research questions show that our work on knowledge discovery in semantically distributed data sites is only at the beginning. However, we convinced that this problem is important for the KDD research community and end users alike, and we hope to inspire further research.

References

R. Agrawal, T. Imielinski, A. Swami. Mining Association Rules between Sets of Items in Large Databases. in: Buneman, P. & Jajodia, S. (Ed.) *Proceedings of the ACM SIGMOD Conference on Management of Data*. May, 26-28, Washington DC, USA, SIGMOD Record 22(2), pp. 207-216. 1993

R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. Proceedings of the 20th International Conference on Very Large Databases (VLDB '94)},1994,

C. Borgelt, R. Kruse, G. Lindner: Lernen von probabilistischen Netzen aus Daten: Theorie und Anwendung, KI-Zeitschrift, 1998.

P. Chan and S. Stolfo. Meta-learning for multistrategy and parallel learning. In Proceeding of the Second International Work on Multistrategy Learning, pages 150-165, 1993.

D. W. Cheung, V. T. Ng, A. W. Fu, and Y. J. Fu. Efficient mining of association rules in distributed databases. IEEE Transactions on Knowledge and Data Engineering, 8:911-922, December 1996.

Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. Machine Learning, 9, 309-347.

T. Dietterich. Ensemble methods in machine learning. *Proc. of the First International Workshop on Multiple Classifier Systems. pp 1- 15*. 2000

J. Hipp, G. Lindner. Analyzing warranty claims of automobiles. an application description following the CRISP-DM data mining process. In Proceedings of 5th International Computer Science Conference (ICSC '99), pages 31-40, Hong Kong, China. 1999

H. Kargupta, P. Chan (eds.). Advances in Distributed and Parallel Knowledge Discovery AAAI Press / The MIT Press, 2000

H. Kargupta, B. Park, D. Hershberger, and E. Johnson. Collective data mining: A new perspective toward distributed data mining. In H. Kargupta and P. Chan, editors, Advances in Distributed and Parallel Knowledge Discovery. AAAI/MIT Press, 2000.

W. Kloesgen. Explora: A Multipattern and Multistrategy Discovery Assistant. *In Advances in Knowledge Discovery and Data Mining*. 1996.

W. Lam and A. M. Segre. Distributed data mining of probabilistic knowledge. In Proceedings of the 17th International Conference on Distributed Computing Systems, pages 178-185. IEEE Computer Society Press. 1997.

A. Prodromides, P. Chan, S. Stolfo. Meta-Learning in Distributed Data Mining Systems: Issues and Approaches. In H. Kargupta, P. Chan (eds.). *Advances in Distributed and Parallel Knowledge Discovery*. AAAI Press / The MIT Press, 2000

F. Provost. Distributed Data Mining: Scaling up and Beyond. In H. Kargupta, P. Chan (eds.). *Advances in Distributed and Parallel Knowledge Discovery* AAAI Press / The MIT Press, 2000

S. J. Stolfo, A. L. Prodromidis, S. Tselepis, W. Lee, D. W. Fan, and P. K. Chan. JAM: Java agent to meta-learning over distributed databases. In D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, Proceedings on the Third International Conference on Knowledge Discovery and Data Mining, pages 74-81. AAAI Press.1997.

R. Wirth, T. Reinartz. Detecting Early Indicator Cars in an Automative Database: A Multi-Strategy Approach. in: Simoudis, E., Han, J. & Fayyad, U. (Ed.). Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. August, 2-4, Portland, Oregon. Menlo Park, CA: AAAI Press, pp. 76-81. 1996

S. Wrobel. An algorithm for multi-relational discovery of subgroups. *Proc. First International Conference on Principles of Data Mining and Knowledge Discovery*. 1997

M. J. Zaki. M. Ogihara, S. Parthasarathy, W. Li. Parallel data mining for association rules on shared memory multi-processors. In *Supercomputing* '96, 1996.