

Learning in and for Real-Time Decision Support and Diagnosis Systems

Michael Borth

DaimlerChrysler - Research and Technology RIC/AM
Postfach 2360, 89013 Ulm, F. R. Germany
Phone: +49 731505-2464, Fax: -4210
Michael.Borth@DaimlerChrysler.com

Abstract

Adaptation to changes over time as well as typical modes of usage, model calibration, and learning of preferences are some of the examples in which expert systems for real-time diagnosis and decision support benefit from learning. Complex mobile systems with these functionalities, such as those produced by DaimlerChrysler, thus require efficient real-time learning methods. Focusing on parameter and structure learning for Bayesian networks, we introduce basic realizations, illustrate their prospects and limitations, and, as the main topic of this paper, deduce the areas which will, from our point of view, most benefit from future research.

Introduction

Given the well-known advantages of Bayesian networks for the real-time diagnosis, prediction, or analysis of dynamic systems, we evaluate their use in next-generation systems with these functionalities at DaimlerChrysler.

Today, we have the means to generate Bayesian networks for the modeling of complex mechatronic systems, user-profiles, behavior, and short- and long-term goals. Using hardware or software realizations, we can implement these networks in ways that satisfy our requirements for inference under uncertainty in real-time expert systems.

Although the networks perform their task well, they benefit from additional adaptation, either by adjustment of the network parameters or by structure learning to discover dependencies. And for these adaptations, we utilize the operational data of the systems themselves.

Real-Time Learning in Mechatronic Systems

Technically, we distinguish between two scenarios for learning in real-time systems: continuous online learning and triggered batch learning. Figure 1 depicts the flow of data and information during learning in both scenarios.

Continuous online learning is realized on a subset of the operational data. It processes only the current state of the system. Triggered batch learning uses a first-in-first-out short-term memory as data source, allowing us to look back at all the relevant data if an event occurs that necessitates learning over a longer time-span. It can be triggered by a monitoring system, e.g. if an error occurs, or by a request for data mining received via a telematics unit. The parameters of both the learning algorithms and the preprocessing unit are determined by the triggering event.

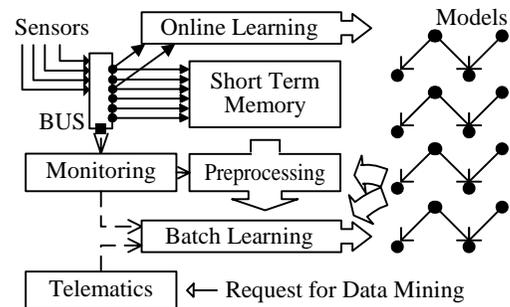


Figure 1: Data flows for online and batch learning

We restrict learning to dependencies that are expected to change over time (this includes the discovery of new ones). This restriction is necessary if we are to cope with the quantities of data generated in modern mechatronic systems for hundreds or thousands of variables many times a second.

The following sections describe the various methods and learning tasks we realize with both scenarios.

Model Calibration

Model calibration, i.e. the adaptation of a model to the specifics of a given system, is necessary to adjust a generic model to differences between systems of the same class that result from varieties in the systems' configurations and from tolerances of the sensor input. In the majority of cases, a short period of online parameter learning for a reduced set of variables is sufficient for this task.

The adaptation and revision of well-defined probability distributions during parameter learning is sufficiently fast for real-time scenarios, especially if limited to the a priori probabilities of variables that depend on no others. While we see little room for actually improving the learning algorithm, the process as a whole can be enhanced in two ways. First, sensitivity analyses may be employed to determine whether (and for which variables) additional learning will lead to improved network performance. This results in more efficient learning, thus reducing effort. Second, more intelligent ways to determine the data needed for learning to guarantee optimal network performance may be used. This is necessary, since experience in data mining shows that learning may result in inferior models that are ill prepared to handle unlikely, but critical situations, even if the input data sets represent the real-world data adequately.

Adaptation to Changes over Time

The second and more complicated learning task for online learning is the adaptation of the Bayesian networks to changes over time. For example, the likelihood of defects is influenced by component wearout, which in turn depends on usage statistics and aging. Usage statistics and similar factors are accumulations of observable evidence whose real-time processing poses no challenge. Component wearout, however, is an example for a class of variables that while not observable, are critical for diagnosis. In addition, influencing factors, e.g. operational temperature, may be unobservable or even unknown. Thus, our task here is parameter learning with hidden variables.

State-of-the-art algorithms allow for real-time learning, given the computational power required by the high complexity. Yet, we have to balance the advantages against the costs incurred. In the near future, we expect such computations to remain restricted to high-end systems and key functions. While we do not anticipate significant improvements of the algorithms in comparison to the increasing computer performance, the latter will enable us to use these methods where needed in the future.

Learning from Infrequent Events

Learning from infrequent events and other tasks necessitates the discovery of previously unknown dependencies between variables, possibly even over multiple time-steps. For example, the discovery of influence factors for temporal faults that occur only infrequently requires learning that they only happen under specific circumstances and often depend on the recent state history of the system. It is not known in advance which variables describe (or hint of) such circumstances. Another example is learning of user-preferences in atypical situations in which a user expects support from an assistance systems, because (s)he (and thus also the expert system) is unfamiliar with the situation.

However, we do not perceive that present algorithms for the resulting task of structural learning with hidden variables are sufficiently fast for real-time online learning targeting at all potentially relevant variables. Instead, we initiate a batch learning process if a trigger event occurs. This process learns a model about the current situation and its history. The model is then refined whenever a similar trigger event occurs. Over time, we may or may not learn what is typical about the situation. This approach, as any other, is not guaranteed to solve the issue that the interesting learning examples are very sparse, or, more formally, that we face a highly unbalanced class distribution - and thus a difficult learning task even without real-time constraints.

Feedback of Model Information into Learning

The more complex of our learning tasks require the processing of information encoded in the Bayesian networks in addition to the input from sensor data. This feedback of model information into learning differs from standard input: Not evidence, i.e. information that a variable has a certain value, but probability distributions, in certain circumstances

even information about the existence of dependencies, form the input. Additional mechanisms to control the learning process are a partial solution for such situation, but an adaptation of the learning algorithms is also necessary. Learning of a model with attention to the current state of the model is not restricted to real-time scenarios, but we consider it especially important not to disregard the high-level information available in the networks under such constraints.

Conclusion and Further Research

The learning tasks we encounter at DaimlerChrysler cover a wide range of real-world applications. They are not restricted to our field of expertise and solutions should be transferable to other areas without much effort. Therefore, we hope that this paper will initiate a fruitful discussion and additional research on the topics we have illustrated.

With today's algorithms for inference and parameter learning, we consider Bayesian networks to be a sensible choice for self-adapting expert systems. Improvements of the algorithms are welcome - and sometimes necessary. But given the underlying complexity of the adaptation tasks, we do not expect any breakthroughs. Instead, we have to look at methods to optimize the algorithms for hardware realizations. Today, dependable and affordable solutions exist for inference. As a first step, we need to extend them to learning in fixed situations. Later, we have to add the flexibility needed to cope with the continuously changing situations and tasks we face for modern mobile systems.

Additionally, the network compilation requires further research: modern network compilers use heuristics to transform networks into an 'efficient' structure. However, it is seldom taken into account that definitions of efficiency depend on the context. For example, smaller structures are needed for hardware realizations of common components (they are cheaper), but, for critical control systems, this may not matter in comparison to speed. Thus, we believe that the industrial implementation of Bayesian networks will benefit from the development of open compiler systems which allow for an optimization of networks to specific requirements, from design guidelines specifying the best practice for common goals, and case studies, which describe efforts and benefits of such optimizations.

Further on, real-time machine learning reconfirms a data mining lesson: an intelligent approach, based on domain knowledge and understanding of machine learning that consists not only of modeling but also of business and data understanding, data preparation, and model deployment, is more valuable with respect to the quality of results than improvements of algorithms will ever be.

Acknowledgements

This paper describes work of the author and several other researchers at DaimlerChrysler. Special acknowledgements go to Rüdiger Wirth (Research Machine Learning and Information Mining), Hermann von Hasseln (Research Electronics and Mechatronics), and Harald Renninger (Research Electric/Electronics Architecture & Diagnosis).