Probabilistic Fault Diagnosis

with Automotive Applications

Anna Pernestål

Probabilistic Fault Diagnosis
with Automotive Applications

annap@isy.liu.se
http://www.vehicular.isy.liu.se
Department of Electrical Engineering,
Linköping University,
SE–581 83 Linköping,
Sweden.

ISBN 978-91-7393-493-0
ISSN 0345-7524

Printed by LiU-Tryck, Linköping, Sweden 2009

To my parents
Kjell and Eva

Abstract

The aim of this thesis is to contribute to improved diagnosis of automotive vehicles. The work is driven by case studies, where problems and challenges are identified. To solve these problems, theoretically sound and general methods are developed. The methods are then applied to the real world systems.

To fulfill performance requirements automotive vehicles are becoming increasingly complex products. This makes them more difficult to diagnose. At the same time, the requirements on the diagnosis itself are steadily increasing. Environmental legislation requires that smaller deviations from specified operation must be detected earlier. More accurate diagnostic methods can be used to reduce maintenance costs and increase uptime. Improved diagnosis can also reduce safety risks related to vehicle operation.

Fault diagnosis is the task of identifying possible faults given current observations from the systems. To do this, the internal relations between observations and faults must be identified. In complex systems, such as automotive vehicles, finding these relations is a most challenging problem due to several sources of uncertainty. Observations from the system are often hidden in considerable levels of noise. The systems are complicated to model both since they are complex and since they are operated in continuously changing surroundings. Furthermore, since faults typically are rare, and sometimes never described, it is often difficult to get hold of enough data to learn the relations from.

Due to the several sources of uncertainty in fault diagnosis of automotive systems, a probabilistic approach is used, both to find the internal relations, and to identify the faults possibly present in the system given the current observations. To do this successfully, all available information is integrated in the computations.

Both on-board and off-board diagnosis are considered. The two tasks may seem different in nature: on-board diagnosis is performed without human integration, while the off-board diagnosis is mainly based on the interactivity with a mechanic. On the other hand, both tasks regard the same vehicle, and information from the on-board diagnosis system may be useful also for off-board diagnosis. The probabilistic methods are general, and it is natural to consider both tasks.

The thesis contributes in three main areas. First, in Paper 1 and 2, methods are developed for combining training data and expert knowledge of different kinds to compute probabilities for faults. These methods are primarily developed with on-board diagnosis in mind, but are also applicable to off-board diagnosis. The methods are general, and can be used not only in diagnosis of technical system, but also in many other applications, including medical diagnosis and econometrics, where both data and expert knowledge are present.

The second area concerns inference in off-board diagnosis and troubleshooting, and the contribution consists in the methods developed in Paper 3 and 4.

The methods handle probability computations in systems subject to external interventions, and in particular systems that include both instantaneous and non-instantaneous dependencies. They are based on the theory of Bayesian networks, and include event-driven non-stationary dynamic Bayesian networks (nsDBN) and an efficient inference algorithm for troubleshooting based on static Bayesian networks. The framework of nsDBN event-driven nsDBN is applicable to all kinds of problems concerning inference under external interventions.

The third contribution area is Bayesian learning from data in the diagnosis application. The contribution is the comparison and evaluation of five Bayesian methods for learning in fault diagnosis in Paper 5. The special challenges in diagnosis related to learning from data are considered. It is shown how the five methods should be tailored to be applicable to fault diagnosis problems.

To summarize, the five papers in the thesis have shown how several challenges in automotive diagnosis can be handled by using probabilistic methods. Handling such challenges with probabilistic methods has a great potential. The probabilistic methods provide a framework for utilizing all information available, also if it is in different forms and. The probabilities computed can be combined with decision theoretic methods to determine the appropriate action after the discovery of reduced system functionality due to faults.

Sannolikhetsbaserad Diagnos med Fordonstillämpningar

Den här arbetet har utförts med i första hand ett motiv: att bidra till förbättrad feldiagnos i moderna, hög-automatiserade fordon. För att uppfylla ständigt ökande krav på säkerhet, funktionalitet, tillgänglighet, komfort och minskad miljöpåverkan blir fordon, som till exempel lastbilar och bilar, allt mer komplexa. Detta gör dem också svårare att diagnosticera och felsöka. Samtidigt ökar kraven på precision och hastighet för diagnossystemen. För att fordonen ska uppfylla allt mer krävande miljörelaterade lagkrav behöver allt mindre fel upptäckas tidigare. Noggrannare diagnos ökar tillgängligheten hos fordonet, förkortar verkstadsbesöken, och sänker driftskostnaderna. Bättre diagnos bidrar även till säkrare fordon, för både förare och medtrafikanter.

Diagnos handlar om att hitta fel som är närvarande i ett system genom att använda ett flertal observationer från systemet och relationer mellan dessa. I dagens och morgondagens moderna fordon innebär detta många utmaningar, i synnerhet eftersom de flesta relationer innehåller osäkerheter. Det är utmanande att konstruera noggranna och tillförlitliga fysikaliska modeller av systemen, då de är mycket komplexa och verkar i en omgivning som ständigt förändras när fordonet kör på vägen. Vidare är det ofta svårt att samla data från fordonen för att lära relationer mellan observationer, i synnerhet från feltillstånd, eftersom fel typiskt är ovanliga och ibland till och med har okänd effekt på observationerna. Dessutom är beräkningskapaciteten, åtminstone för diagnos som ska utföras ombord på fordonet, ofta begränsad. Detta beror på att de processorer som klarar den utsatta miljön ombord har betydligt sämre prestanda än processorer till exempel i en PC. På verkstaden möts man av svårigheten att felen

i fordonet inte nödvändigtvis är synligt när fordonet är stilla. Till exempel är det svårt att upptäcka problem med bromsarna när inte bromsarna används.

Flera av utmaningarna inom fordonsdiagnos är relaterade till osäkerheter och otillräcklig information. Därför antas ett sannolikhetsbaserat förhållningssätt i den här avhandlingen, både när det gäller att hitta relationerna mellan observationerna, och för att detektera fel. Målet är att beräkna sannolikheterna att lika fel är närvarande. För att lyckas med detta är det viktigt att all tillgänglig information används i beräkningarna.

I avhandlingen betraktas både diagnos utförd ombord på fordonet och diagnos gjord på verkstad. Diagnos ombord och verkstadsdiagnos kan förefalla vara två helt olika problem. Ombord görs diagnosen automatiskt i styrsystemet och (i de flesta fall) helt utan inblandning av människor, till skillnad från diagnos på verkstäder som i första hand utförs av mekanikern, stöttad av ett felsökningsverktyg. Å andra sidan gäller diagnosen samma fordon, och information från diagnosen i styrsystemet ombord kan vara till stor hjälp under felsökningen på verkstaden. Inom det ramverk för diagnos, baserat på sannolikhetsteori, som används och utvecklas i den här avhandlingen, är metoderna generella och kan appliceras på diagnos både ombord och på verkstaden. Därför blir det naturligt att betrakta båda typerna av diagnos.

Den här avhandlingen bidrar i första hand inom tre områden. Det första området är metoder för att kombinera olika typer av information i sannolikhetsberäkningar. I artiklarna 1 och 2 har metoder utvecklats för att kombinera träningsdata och expertkunskap av olika typer. Metoderna är generella och kan inte bara användas inom diagnos, utan även inom många fält, till exempel medicinsk diagnos och ekonomisk modellering. Metoderna i artiklarna 1 och 2 har i första hand utvecklats med avseende på diagnos ombord, men kan självklart även användas inom verkstadsdiagnos.

Det andra området avhandlingen bidrar till är inom modellering och sannolikhetsberäkningar för felsökning på verkstäder. Artiklarna 3 och 4 beskriver sådana metoder. Den största utmaningen i felsökning är att hantera yttre påverkan på systemet. Till exempel, när fordonet repareras förändras systemet och de beroenden som finns mellan komponenter förändras och försvinner. Metoderna som utvecklats i artiklarna 3 och 4 är baserade på Bayesianska nätverk, och innefattar bland annat ett nytt ramverk för händelse-styrda icke-stationära dynamiska Bayesianska nätverk och en effektiv men enkel algoritm för att kunna använda vanliga statiska Bayesianska nätverk i modellering för felsökning. Ramverket för händelse-styrda icke-stationära dynamiska Bayesianska nätverk är inte enbart användbara inom felsökning, utan och kan användas i många frågeställningar där sannolikhetsberäkningar ska göras i system som utsätts för yttre påverkan.

Det tredje bidraget, presenterat i artikel 5, är en jämförelse och utvärdering av olika metoder lära relationer mellan observationer och fel från träningsdata.

Att lära från data för diagnos ställer särskilda krav på algoritmerna som används, och i artikel 5 har ett fem olika metoder anpassats till diagnos-problemet och deras prestanda har jämförts.

Genom hela avhandlingen har arbetet drivits av fallstudier av delsystem i en modern lastbil, där olika problem och svårigheter har identifierats. Teoretiskt sunda och generella metoder har utvecklats för att lösa dessa problem. Metoderna har sedan applicerats på de riktiga systemen i lastbilen.

Preface

I believe searching faults is like a detective's work. We observe the system, discuss the hidden relations, using whatever we know about the system, and draw conclusions about whether there are faults present and, if so, which faults. Therefore, searching faults and doing diagnostic work is about understanding relations between observations and different faults, and to distinguish the relevant information in the observations. To design a diagnosis system, we have to find the relations. To perform diagnostic work, we have to reason using the relations and the current observations.

There are several different methods for learning the hidden relations in systems to diagnose: building models, using data, applying expert systems, and so on. However, digging deeper into the problem designing a diagnosis system, we notice that the available information is (often) not sufficient to exactly determine if there are faults present, nor to distinguish between them. We are left with a bunch of possible explanations.

This fact leads into the field of probability theory. When dealing with probabilities, and in particular probabilities about "real-world" events, such as "what is the probability that this truck is fault free?", one need to know what "probability" is.

So, what is probability? Before beginning the work with this thesis, I would have said something like "Well, the probability is the relative frequency. I suppose." However, I must confess, I had some problems with this interpretation. First, even if fault F is present in 1 out of 100 trucks, i.e. has relative frequency 0.01, what is the probability that the fault is present in *this* particular truck?

Second, if a person I trust tells me that this truck is fault free, what is the probability that this the truck is fault free then? It is reasonable that it depends on how *much* I trust the person?

My problems with the interpretation of probability are, at least philosophically, solved through inspiring and interesting discussions with Mikael Sternard and Mathias Johansson at the Signals and Systems group at Uppsala University five years ago. They introduced me to E. T. Jaynes' book *Probability – the Logic of Science* on probability as an extension to logic. According to Jaynes, probability is a property of the spectator and his state of knowledge rather than a "physical" property of the object. This gave me an understanding of probability as a measure of belief that has made this thesis possible. Without Mikael and Mathias it is highly probable that this thesis had been something completely different.

One of the most important persons during the work with this thesis has been my supervisor Dr. Mattias Nyberg. He has supported me through this work by pushing my ideas further, and efficiently puncturing my bad ideas. He has always new questions coming up, and new ideas about how the world and the work is. It has been an intellectual challenge to work with Mattias - and I love challenges.

This thesis has been performed as a collaborative industrial research project between Scania CV AB in Södertälje and the division of Vehicular Systems, Department of Electrical Engineering, Linköping University. I thank my managers at Scania for supporting this work and making it financially possible. Thanks to Prof. Lars Nielsen, for letting me join the Vehicular Systems group in Linköping, and to the people at the group, and in particular at the diagnosis group of Vehicular Systems, for the interesting discussions and for broadening my perspective on diagnosis (and many other things).

Other persons that have been more important for this work are my co-supervisor Dr. Jose M. Peña, with his knowledge on Bayesian networks; Dr. Nils-Gunnar Vågstedt and Hans Ivendahl at Scania, with their encouragement, and "real-world related questions" that have helped me to focus on the real problems; and Prof. Petri Myllymäki and Hannes Wettig at the CoSCo group at Helsinki University for hosting me and introducing me to learning methods.

Carl Svärd, Håkan Warnquist, and Dr. Tony Lindgren have proof-read parts of this thesis. Your comments have been invaluable.

A special thank to Dr. Erik Frisk for his support on LaTeX, his never-ending interest, and his clever comments and questions.

To Support(er) Petter Lindh, for his infectious harmony, and his thoughtful comments, always given to me with excellent timing and content.

Many people know that I am addicted to long-distance running, and I think that the work with this thesis has been much like running a marathon race. A marathon is a challenge that, during the race, is sometimes simply fun, some-

times painful and heavy, often exhausting – but, the whole way through, a great pleasure! I will end this marathon with thanking my supporters that have helped me, encouraged me, and supported me through this marathon: my friends, my grandparents, and my wonderful family Karin, Kjell, Eva, and Johan.

Anna Pernestål
Linköping 2009

Contents

Part I

Introduction

1

Introduction

You insist that there is something a machine cannot do. If you tell me precisely what it is that a machine cannot do, then I can always make a machine that does just that!

J. von Neumann, 1948

1.1 Background

1.1.1 Why Automotive Diagnosis?

To meet steadily increasing requirements on performance, safety, and decreased environmental impact, modern automotive vehicles are becoming increasingly complex products. For example, functions are developed for active safety systems, for exhaust gas after-treatment, and to optimize fuel economy. The functions typically integrate mechanics, chemical processes, hydraulics, and electric components, as well as electronic control units (ECUs) and software. The number of ECUs is steadily increasing to satisfy requirements on increased functionality. As an example, during the last fifteen years the number of ECUs in an Scania heavy truck has increased from about five fifteen years ago to about 35-40 in modern trucks of today.

The complexity and increased functionality of modern vehicles make them more challenging to monitor, diagnose, and troubleshoot. At the same time the requirements on the diagnosis system itself are increasing As depicted in

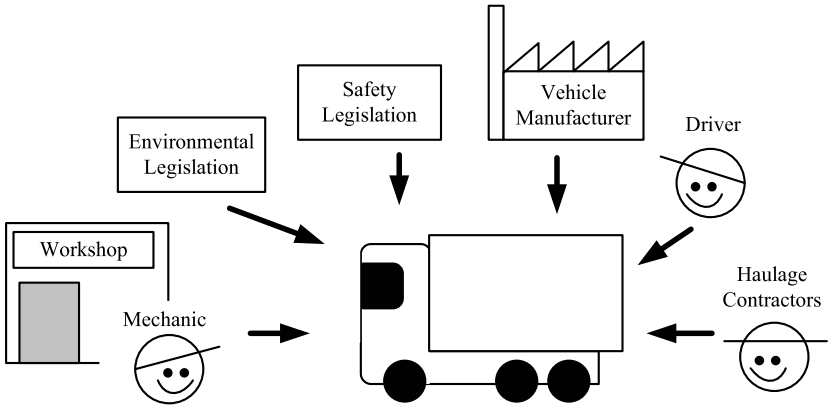


Figure 1.1: There are several interesents with requirements on the diagnosis system of an automotive vehicle.

Figure 1.1, there are several interesents having requirements on the diagnosis system in a heavy truck. At the workshop, the mechanic needs support to be able to perform fast and efficient troubleshooting and repair of the complex automotive vehicles. To fulfill demanding environmental legislation, faults that increase exhaust emissions must be detected within specified times, and safety legislation regulates faults related to safety issues. The manufacturer needs a diagnosis system that is easily configured during development of new products, and that can be used also in the early phases of product testing. A powerful diagnosis system is also an important factor for manufacturers of automotive vehicles in the competition for customers, and will continue to be so in the future. For the driver, the diagnosis system should reduce safety risks without producing any unexpected behavior of the truck, nor any annoying false alarms. For haulage contractors, increased uptime and reduced service and maintenance costs are important. This can be achieved with an accurate and efficient diagnosis system.

1.1.2 Diagnosis is a Challenge

Fault diagnosis is about finding faults that possibly are present in the system by using numerous observations and their internal relations. The internal relations can be described by different types of models of the system and the faults. However, in complex systems, such as automotive vehicles of today and tomorrow, finding the internal relations and building models is a most challenging task, since the relations often are hidden and may include uncertainty. Building accurate physical models of the automotive systems is complicated, both due to the complexity of the systems and since the systems are operated

in continuously changing surroundings. In addition, in particular considering heavy trucks and buses, many vehicles are rebuilt or reconfigured after leaving the factory to satisfy customers' specific needs. Reconfigurations can for example include containers with refrigerators for food transport, changed in-take air systems for trucks operating in deserts, external systems for handling timber, or changed rear axle gear ratio. These reconfigurations lead to that the knowledge about the actual configuration of the vehicle in the on-board diagnosis system in the ECU is limited. Uncertainty is further increased by measurement noise in sensors, and by the dispersion in quality in the sensor populations.

In automotive diagnosis, collecting data to learn from is often difficult, mainly since faults are rare. One alternative is to implement faults and collect data. However, since there are many different faults, and some are difficult, or even impossible, to implement, there will most often only be a limited data available from a small subset of faults that should be diagnosed. In particular, there will typically only be data from single faults, but the diagnosis system should also handle multiple faults. Moreover, there may be faults that causes abnormal behavior but that are previously unseen.

On-board the vehicle, diagnosis is performed in ECUs, where the hardware capacity in terms of CPU power and data storage is limited. Off-board, at the workshop, the hardware capacity is less limited. On the other hand, there can be faults that are present in the vehicle but that are not excited while the vehicle is at the workshop.

1.1.3 Approaches to Diagnosis

One common and efficient approach to diagnosis is to use models of the system and apply model based diagnosis (MBD). The models can be of different types. Each of the model types have different advantages and drawbacks in the automotive application.

One important class of models used for diagnosis is physical models, as for example in [Peischl and Wotawa, 2003, Lucas, 2001, Hamscher et al., 1992, Korbicz et al., 2004, Gertler, 1998]. However, as described in the previous section, accurate modeling of automotive vehicles is difficult due to several sources of uncertainty. Other diagnosis methods are based on models learned from data, as for example the ones in [Gustafsson, 2001, Russell et al., 2000, Basseville and Nikiforov, 1993]. The collection of data for diagnosing automotive vehicles is associated with two main problems. First, to distinguish faults with data driven techniques, data from both the fault free case as well as from fault situations is needed. Data from the fault free case can typically be collected by running and observing the system. Data from faults is, on the other hand, difficult to collect from observing the system since faults are rare. Second, the diagnosis system should work when the product is newly released to the market, but at this point the amount of data is often limited. In addition, each new release

of a product needs a new set of data, even when the differences from previous releases only are small.

The uncertainties described in the section above makes it difficult, or even impossible, to determine exactly which fault that is present in the vehicle. Many diagnosis algorithms, for example those based on the General Diagnosis Engine (GDE) [de Kleer, 1992] and its extension Sherlock [de Kleer and Williams, 1992] or Reiter's method based on first principles logic [Reiter, 1992], determines a list of all faults that can possibly explain the current behavior of the system. With uncertainty in models and measurements, this list may be very long, since faults can not be excluded with certainty. In GDE, Sherlock, and Reiter's methods these lists are focused on more probable faults, mainly in the sense that explanations with a small number of faulty components are preferred before explanations with a larger number of faulty components. In this thesis, we handle the uncertainties by taking a probabilistic approach, and compute the probabilities for the faults and combination of faults, given all available information. In the probabilistic approach, faults that are impossible are assigned probability zero, and possible faults are ranked after their probability. In addition, having computed the probabilities for faults, we can apply a decision-theoretic approach, where the probabilities are combined with a loss function to determine the counter action to perform. The concept of combining probabilities with loss functions can be used both for on-board and off-board diagnosis, but with different loss functions.

1.2 Problem Formulation

The main objective in this thesis is to contribute to improved diagnosis of automotive vehicles. We let the work be driven by case studies of real applications, where challenges and problems are identified. Methods for solving the identified problems are developed, and applied to the real systems. Fault diagnosis is a challenging and complicated task, and although the tasks of diagnosing different systems or subsystems are similar, there are also differences, for example in the type of background knowledge available. We strive for making the diagnosis methods theoretically sound and general. The soundness of the methods makes it easier to track and understand the meaning of their output and to guarantee their performance. Moreover, development engineers can tailor the general methods to suite their particular application.

We consider both on-board and off-board diagnosis of automotive vehicles. The two tasks may seem different in nature. On-board diagnosis is performed in the automotive on-board control system during operation of the vehicle, mostly without human integration. Off-board diagnosis is performed by a mechanic supported by a troubleshooting tool. In the troubleshooting tool, diagnosis is based on the possibility of human interaction with the system. On the other

hand, both on- and off-board diagnosis regard the same vehicle, and models used in diagnosis rely on the same internal relations in, or models of, the system.

Within the probabilistic framework used in this thesis, the main objective is to compute the probability distribution for faults, or *system status*, given all information available:

$$p(\textit{system status}|\textit{all available information}). \tag{1.1}$$

This probability can then be combined with decision theory to determine the appropriate action; for example the best on-board control strategy, the best troubleshooting action, whether to set of an alarm or not, etc. The probability (1.1) is used both on-board and off-board. "All available information" can be divided into three main parts: expert knowledge about the system, data, and current observations. The expert knowledge and the data are the same in both on- and off-board diagnosis of the same vehicle, and therefore it is natural to consider both tasks in the thesis. However, since different kinds of observations are available for on-board diagnosis during operation and at off-board at the workshop, different subparts of expert knowledge and data may play different roles. This also means that information stored from the on-board diagnosis may contribute to improved off-board diagnosis, and vice versa.

The computation of the probability (1.1) is central in this thesis. We consider different kinds of information, or knowledge, and use different computation approaches. In particular, we focus on the following questions:

- How do standard methods for learning from data perform in the computation of (1.1)?
- Which are the main issues regarding the training data available for diagnosis?
- In the computation of (1.1), the different pieces of information are to be combined. The different information pieces can be of widely different types, and include for example dynamical physical models, state machines, fault models, structural knowledge about fault effects, experimental and observational data, function specifications. How should these different kinds of information be integrated in the computations?
- To compute (1.1), dependency relations between different subparts of the diagnosed system are used. In probabilistic terms, the dependency relations represent information flow. However, the physical relation that caused the dependency may not be present at the time the relation is used. For example, at the work shop it can be observed that oil has leaked out during operation of the system, although there is no oil leaking out when the system is at rest. In particular, during off-board diagnosis, what are the effects of these different kinds of dependencies?

- During off-board diagnosis and troubleshooting at workshops, "all information available" includes knowledge about that parts of the system have been repaired. The repairs are external interventions that change the dependency structure of the system. Therefore, one important question is: how should external interventions be handled in the computation of (1.1)?
- In on-board diagnosis, hardware capacities are limited, and in off-board diagnosis fast computations are crucial to reduce troubleshooting and repair time. Therefore, one important question is thus: how to compute the probability (1.1) as efficiently as possible?

2

Contributions

We balance probabilities and choose the most likely. It is the scientific use of the imagination.

Sherlock Holmes, in "The Hound of the Baskervilles", 1902

2.1 Thesis Overview

Besides this introductory part, Part I, this thesis consists of three parts: an introduction and brief survey of probabilistic methods for diagnosis, five appended papers, and conclusions. An overview of the three parts, and relations between the papers and chapters is shown in Figure 2.1.

Part II is an introductory survey of probability, diagnosis, and in particular probabilistic methods for diagnosis. It constitutes, together with the current Part I, the bottom layers in Figure 2.1. In Chapter 3, a brief introduction to Bayesian probability is given. Rather than a being a reference on probability theory presenting computation rules, it is intended as a discussion of interpretations of probability. In particular the interpretation used in this thesis is presented. In Chapter 4 a brief survey of previous works on model-based diagnosis, and in particular probabilistic diagnosis.

Part III is the main part of this thesis, and consists of the five appended papers. In all five papers there are both application-related and theoretical contributions. The theoretical contributions are in the fields of learning, modeling and inference. As depicted in Figure 2.1, Papers 1, 2, and 5 contribute to the

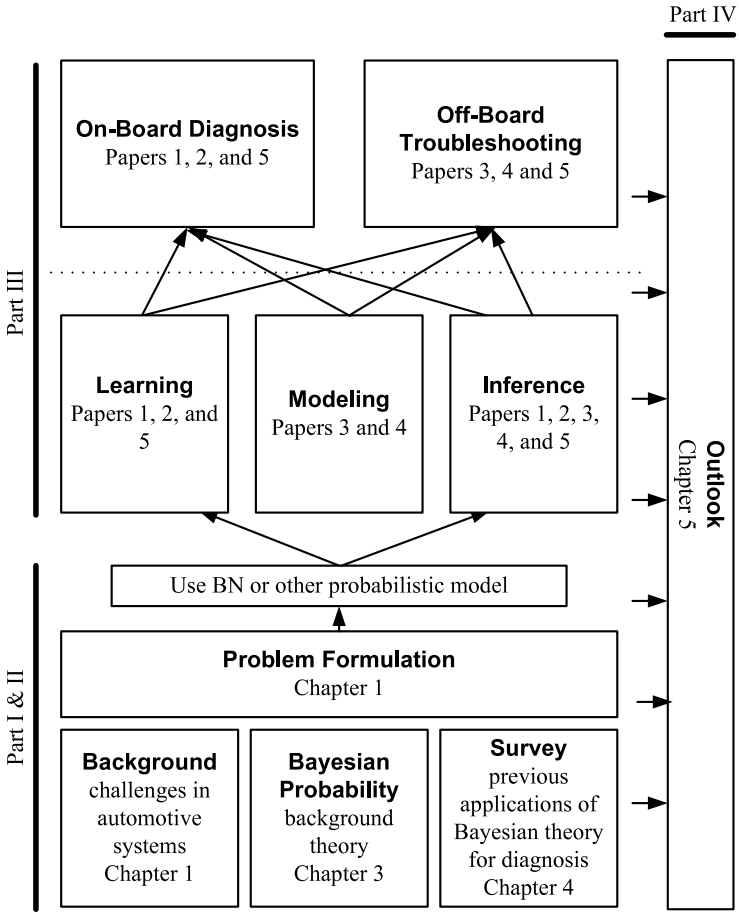


Figure 2.1: Overview of the thesis.

theory of learning, while Papers 3 and 4 consider learning. All five papers have theoretical contributions in the field of inference. In the application-related view, Papers 1 and 2 have a clear focus on on-board diagnosis, while Papers 3 and 4 focus on off-board troubleshooting. In Paper 5 theoretical methods are handled that are applicable to both on- and off-board diagnosis.

In Part IV, a conclusion of the work and the results in the thesis are presented. Moreover, an outlook is provided, discussing future challenges and applications of probabilistic diagnosis in automotive systems.

2.2 Appended Papers – Summary and Contributions

In this section we give an overview of the appended papers, together with a brief summary of each of the papers. For each paper, we also present the contributions, both the theoretical contributions related to development of new methods, and the application-related contributions related to diagnosis of real automotive vehicles.

2.2.1 Paper 1 - Data and Process Knowledge

Anna Pernestål and Mattias Nyberg. (2008). Bayesian Fault Diagnosis for Automotive Engines by Combining Data and Process Knowledge. Submitted to *IEEE Transactions on Systems, Man, and Cybernetics part A*.

Paper 1 is based on the publication:

- Anna Pernestål and Mattias Nyberg. (2007). Probabilistic Fault Diagnosis Based on Incomplete Data. In *Proceedings of the European Control Conference (ECC 2007)*, Kos, Greece.

Summary

The objective is to develop a diagnosis method that computes probabilities of faults, and that is applicable to real automotive systems. A careful application study is performed, and requirements on the diagnosis system are listed.

The diagnosis method should compute the probabilities for faults, using all available information. The case study has shown that the available information comprise several types of information: training data; different kinds of monitoring functions, such as diagnostic tests or residuals; and sensor readings. The training data available is typically limited in amount. Furthermore, the training data is often experimental, i.e. collected after first actively implementing faults, instead of simply observing the system and wait for faults to appear. For many automotive systems there are physical models available of the system, but they are typically not detailed enough to rely on alone in fault diagnosis. Finally,

the computational burden should be kept small to meet hardware capacity limitations of on-board ECU processors.

A method for computing the probabilities of faults given both the physical models and the (limited amount of) training data is developed. The method is a combination of two previous types of methods – consistency based methods using the Fault Signature Matrix (FSM) such as Sherlock [de Kleer and Williams, 1992] and structured hypothesis testing [Nyberg, 2000], and standard probabilistic methods using training data only, see for example [Heckerman et al., 1995]. In an application to the task of diagnosing the gas flow of a heavy truck diesel engine, the new method is illustrated on real world data.

In the paper it is also discussed how the new, combined method relates to these previous methods for diagnosis, and that the diagnosis result always is at least as good as using one of the previous methods.

Contributions

- The detailed investigation of the automotive diagnosis problem.
- The translation of physical characteristics of the diagnosed process to assumptions in the probability computations.
- The method for combining training data and expert knowledge in terms of an FSM in computations of probabilities of faults.
- The application of the new method to the diagnosis of a real world automotive diesel engine.
- The investigation of the new method's relation to previous works such as Sherlock [de Kleer and Williams, 1992], structured hypotheses testing [Nyberg, 2000], model-based probabilistic methods, and Bayesian networks.

2.2.2 Paper 2 - Data and Likelihood Constraints

Anna Pernestål and Mattias Nyberg. (2007). Bayesian Inference by Combining Training Data and Background Knowledge Expressed as Likelihood Constraints. Submitted to *International Journal of Approximate Reasoning*.

Paper 2 is based on the publication:

- Anna Pernestål and Mattias Nyberg. (2007). Using Prior Information in Bayesian Classification - with Application to Fault Diagnosis. In *27th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (MaxEnt 2007)*, Albany, USA.

Summary

A new method is developed for learning the posterior probability distribution of a class variable C given an observation vector $\mathbf{x} = (x_1, \ldots, x_n)$ and background information $\mathbf{i}$ consisting of a combination of training data and expert knowledge in terms of likelihood constraints. Likelihood constraints are constraints on linear combinations on the parameters in the distributions $p(x_i|C, \mathbf{i})$. The likelihood constraints are very general, and can be used to express several types of expert knowledge, such as explicit knowledge about certain values of the parameters in the probability computations, or knowledge about the values of linear combinations of parameters. Also, constraints such as "variable X_i has the same, but unknown, distribution given $C = c_1$ and $C = c_2$" can be expressed using likelihood constraints.

Likelihood constraints appear naturally in many different kinds of applications, such as medical and technical diagnosis and econometrics. In particular, the constraints in probability computations considered in the previous papers [Boutilier et al., 1996] and [Jaeger, 2004] are special cases of the likelihood constraints considered here.

In the paper, the derivation of the new method is shown in detail. The method leads to multidimensional integrals that do not have any closed form solutions in general. In the paper, an approximate solution method based on Laplace approximation is proposed. All the computations are illustrated in detail on two examples, of which one is a diagnosis task.

Contributions

- The method for integration of expert knowledge in terms of likelihood constraints and training data in probability computations.
- The translation of constraints in general terms into likelihood constraints.
- The application of the new method to the diagnosis problem.

2.2.3 Paper 3 - Non-Stationary Dynamic Bayesian Networks

Anna Pernestål and Mattias Nyberg. (2009). Non-Stationary Dynamic Bayesian Networks in Modeling of Troubleshooting Processes. Submitted to *International Journal of Approximate Reasoning*.

Paper 3 is partly based on the publication:

- Anna Pernestål, Håkan Warnquist, and Mattias Nyberg. (2009). Modeling and Troubleshooting with Interventions Applied to an Auxiliary Truck Braking System. In *Proceedings of 2nd IFAC Workshop on Dependable Control of Discrete Systems (DCDS'09)*, Bari, Italy.

Summary

The task of troubleshooting automotive vehicles is considered, and in particular the computation of probabilities of faults in a process that is subject to external interventions. The task is further complicated by the fact that there is a mixture of two kinds of dependencies that are used in troubleshooting: instantaneous and non-instantaneous. For example, during operation of a vehicle there may be oil leaking out from a pipe through a worn out gasket. When the system is at rest, the oil on the outside of the pipe can be used to identify the leakage, although the oil is not leaking out at rest. If the oil is cleaned up, the system must be operated again in order to verify whether the leakage is still present.

The external interventions changes the dependency structure of in the model: we say that they cause events. To to model processes with both instant and non-instant dependencies and events, the framework of event-driven non-stationary dynamic Bayesian networks (nsDBN) is developed. The framework is general, not only to troubleshooting, but to modeling of all kinds processes where there are events. It is also shown how an event-driven nsDBN is efficiently characterized by an initial Bayesian network (BN), a nominal transition BN, and three sets used to define the events.

Modeling is an artwork, and in the paper we provide guidelines for development engineers to simplify the task. We also describe the troubleshooting problem in the framework of event-driven nsDBN, and illustrate the computations on a typical subsystem of an automotive vehicle.

Contributions

- The general framework of event-driven nsDBN, that facilitates probability computations in systems that are subject to external interventions that affects the dependency structure.
- The formulation of the troubleshooting problem within this framework. This opens for solving troubleshooting problems in the automotive filed, where it is important to handle general dependency structures, multiple faults, and without any simple function verification.
- The illustration of the use of event-driven nsDBN on an automotive example.

2.2.4 Paper 4 - Modeling and Inference for Troubleshooting

Anna Pernestål, Mattias Nyberg, and Håkan Warnquist. (2009). Modeling and Efficient Inference for Troubleshooting Automotive Systems. *Technical Report LiTH-ISY-R-2921.* Department of Electrical Engineering, Linköping University.

Paper 4 is partly based on the publications:

- Anna Pernestål, Håkan Warnquist, and Mattias Nyberg. (2009). Modeling and Troubleshooting with Interventions Applied to an Auxiliary Truck Braking System. In *Proceedings of 2nd IFAC Workshop on Dependable Control of Discrete Systems (DCDS'09)*, Bari, Italy.
- Håkan Warnquist, Anna Pernestål, and Mattias Nyberg. (2009). Anytime Near-Optimal Troubleshooting Applied to an Auxiliary Truck Braking System. In *Proceedings of 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS 2009)*. Barcelona, Spain.

Summary

The objective in this paper is to propose a troubleshooting system that applies to real automotive applications. To do this, a case study of a mechatronic system of an automotive heavy truck, an auxiliary braking system, is performed. Three main issues are identified as important to account for in the troubleshooting system: the need for assembling/disassembling the vehicle during troubleshooting, the difficulty to verify whether the system is fault free, and the need for time efficient inference to reduce waiting time for the mechanic. The first two issues leads to that probabilities need to be computed in a system that is subject to external interventions.

A decision-theoretic approach is used to design a troubleshooting system consisting of two parts: a planner, that suggests the next troubleshooting action; and a diagnoser that supports the planner with probability computations. To compute the probabilities in the diagnoser the framework of event-driven nsDBNs presented in Paper 3 can be used. In the nsDBN probabilities for all ingoing variables can be used, but the diagnoser it is shown to be sufficient to compute conditional probabilities for observations. Therefore, we take off in the nsDBNs, and develop a new method for computing the necessary probabilities in the diagnoser. The method is based on an algorithm that through simple manipulations updates a static BN as events occur. The algorithm is carefully derived and proved in the paper. In the paper we also discuss practical issues related to modeling for troubleshooting.

Contributions

- The development of a troubleshooting system that is applicable to real automotive systems. In particular, assembling/disassembling of the system is possible, and no specific function verification is presumed.
- The detailed case study, and the extensive discussion of practical issues related modeling for troubleshooting.

- The new efficient inference algorithm for troubleshooting, based on an algorithm that updates a static Bayesian network as external interventions occur. In particular, it is proved that the algorithm provides the same probabilities as an nsDBN.

2.2.5 Paper 5 - Comparing Methods for Learning

Anna Pernestål, Hannes Wettig, Tomi Silander, Mattias Nyberg, and Petri Myllymäki. (2009). A Comparison of Bayesian Methods for Learning in Fault Diagnosis. Submitted to *Pattern Recognition Letters*.

Paper 5 is based on the publications:

- Anna Pernestål, Hannes Wettig, Tomi Silander, Mattias Nyberg, and Petri Myllymäki. (2008). A Bayesian Approach to Learning in Fault Isolation. In *Proceedings of 19th International Workshop on Principles of Diagnosis (DX'08)*, Blue Mountains, Australia.

Summary

In this paper, five approaches for learning from data are compared and evaluated on the problem of fault diagnosis and isolation. Based on the five approaches are previously presented in the literature, eight methods where derived. The compared methods are: Direct Inference [Pernestål and Nyberg, 2007], two versions of naive Bayesian networks [Jensen and Nielsen, 2007] with discrete and binary observations respectively, two verisons of general Bayesian networks [Jensen and Nielsen, 2007, Silander and Myllymäki, 2006] with discrete and binary observations respectively, linear regression [Bishop, 2005], logistic regression [Roos et al., 2005], and weighted logistic regression, a version of logistic regression that is developed to handle experimental training data. The methods are tailored to suite the fault diagnosis and isolation problem, and to handle issues in fault diagnosis, such the experimental data and that there are faults from which there is not data.

To evaluate the methods, relevant performance measures are discussed. Finally the methods are compared on data from a real-world automotive diesel engine. Among the compared methods, logistic regression is shown to perform best on this

Contributions

- The application and comparison of eight different Bayesian methods for learning from data, applied to the fault diagnosis problem.
- The investigation of special characteristics of training data in diagnosis, for example that the amount of data often is limited, and that data typically is experimental.

- The tailoring of these methods to suite the fault diagnosis problem, and in particular the unseen fault patterns and the experimental data.

2.3 List of Publications

Here follows a list of publications that are not appended to the thesis, but that constitute an important background work to the appended papers. They are listed in order of publication.

- Anna Pernestål, Mattias Nyberg, and Bo Wahlberg. (2006). A Bayesian Approach to Fault Isolation with Application to Diesel Engine Diagnosis. In *Proceedings of 17th International Workshop on Principles of Diagnosis (DX'06)*, Peñaranda, Spain.

- Anna Pernestål, Mattias Nyberg, and Bo Wahlberg. (2006). A Bayesian Approach to Fault Isolation Structure Estimation and Inference. In *Proceedings of IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS 2006)*, Beijing, China.

- Anna Pernestål. (2006). A Bayesian Method for Fault Identification – a discussion on the Assignment of Priors. In *Reglermöte 2006*, Stockholm, Sweden.

- Anna Pernestål. (2007). *A Bayesian Approach to Fault Isolation with Application To Diesel Engine Diagnosis.* Licentiate Thesis. Toyal Institute of Technology, Stockholm, Sweden.

- Anna Pernestål and Mattias Nyberg. (2007). Using Data and Prior Information in Bayesian Classification. Tech. Report LiTH-ISY-R-2811. Linköping University, Linköping, Sweden.

- Anna Pernestål and Mattias Nyberg. (2007). Probabilistic Fault Diagnosis Based on Incomplete Data. In *Proceedings of the European Control Conference (ECC 2007)*, Kos, Greece.

- Anna Pernestål and Mattias Nyberg. (2007). Using Prior Information in Bayesian Classification - with Application to Fault Diagnosis. In *27th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (MaxEnt 2007)*, Albany, USA.

- Anna Pernestål and Mattias Nyberg. (2007). Experimental and Observational Data in Learning for Bayesian Inference. Tech. Report LiTH-ISY-R-2834. Linköping University, Linköping, Sweden.

- Hannes Wettig, Anna Pernestål, Tomi Silander, and Mattias Nyberg. (2008). A Bayesian Approach to Learning in Fault Isolation. In *Bayesian Modelling Applications Workshop at the 24th Conference on Uncertainty in Artificial Intelligence (UAI 2008)*. Helsinki, Finland.

- Anna Pernestål, Hannes Wettig, Tomi Silander, Mattias Nyberg, and Petri Myllymäki. (2008). A Bayesian Approach to Learning in Fault Isolation. In *Proceedings of 19th International Workshop on Principles of Diagnosis (DX'08)*, Blue Mountains, Australia.

- Anna Pernestål and Mattias Nyberg. (2008). A Bayesian inference under Probability Cosntraints. In *Proceedings of 10th Scandinavian Conference on Artificial Intelligence (SCAI 2008)*, Stockholm, Sweden.

- Anna Pernestål, Håkan Warnquist, and Mattias Nyberg. (2009). Modeling and Troubleshooting with Interventions Applied to an Auxiliary Truck Braking System. In *Proceedings of 2nd IFAC Workshop on Dependable Control of Discrete Systems (DCDS'09)*, Bari, Italy.

- Håkan Warnquist, Anna Pernestål, and Mattias Nyberg. (2009). Anytime Near-Optimal Troubleshooting Applied to an Auxiliary Truck Braking System. In *Proceedings of 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS 2009)*. Barcelona, Spain.

References

[Basseville and Nikiforov, 1993] Basseville, M. and Nikiforov, I. V. (1993). *Detection of Abrupt Changes. Theory and Application*. Prentice Hall, New Jersey.

[Bishop, 2005] Bishop, C. M. (2005). *Neural Networks*. Oxford University Press.

[Boutilier et al., 1996] Boutilier, C., Friedman, N., Goldszmidt, M., and Koller, D. (1996). Context-Specific Independence in Bayesian Networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artifial Intelligence*.

[de Kleer, 1992] de Kleer, J. (1992). Focusing on Probable Diagnosis. *Readings in model-based diagnosis*, pages 131–137.

[de Kleer and Williams, 1992] de Kleer, J. and Williams, B. C. (1992). Diagnosis with Behavioral Modes. In *Readings in Model-based Diagnosis*, pages 124–130, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Gertler, 1998] Gertler, J. J. (1998). *Fault Detection and Diagnosis in Engineering Systems*. Marcel Decker, New York.

[Gustafsson, 2001] Gustafsson, F. (2001). *Adaptive Filtering and Change Detection*. Wiley.

[Hamscher et al., 1992] Hamscher, W., Console, L., and deKleer, J. (1992). *Readings in Model-based Diagnosis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Heckerman et al., 1995] Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20(3):197–243.

[Jaeger, 2004] Jaeger, M. (2004). Probabilistic decision graphs - combining verification and ai techniques for probabilistic inference. *International Journal of Uncertainty, Fuzziness & Knowledge-Based Systems*, 12:19–42.

[Jensen and Nielsen, 2007] Jensen, F. V. and Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs*. Springer.

[Korbicz et al., 2004] Korbicz, J., Koscielny, J. M., Kowalczuk, Z., and Cholewa, W. (2004). *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Springer, Berlin, Germany.

[Lucas, 2001] Lucas, P. J. F. (2001). Bayesian model-based diagnosis. *International Journal of Approximate Reasoning*, 27:99–119.

[Nyberg, 2000] Nyberg, M. (2000). Model Based Fault Diagnosis Using Structured Hypothesis Tests. In *Fault Detection, Supervision and Safety for Technical Processes. IFAC*, Budapest, Hungary.

[Peischl and Wotawa, 2003] Peischl, B. and Wotawa, F. (2003). Model-based diagnosis or reasoning from first principles. *IEEE Intelligent Systems*, 18(3):32–37.

[Pernestål and Nyberg, 2007] Pernestål, A. and Nyberg, M. (2007). Probabilistic Fault Isolation Based on Incomplete Training Data with Application to an Automotive Engine. In *Proceedings of the European Control Conference (ECC 07)*.

[Reiter, 1992] Reiter, R. (1992). A Theory of Diagnosis From First Principles. In *Readings in Model-based Diagnosis*, pages 29–48, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Roos et al., 2005] Roos, T., Wettig, H., Grünwald, P., Myllymäki, P., and Tirri, H. (2005). On Discriminative Bayesian Network Classifiers and Logistic Regression. *Machine Learning*, pages 267–296.

[Russell et al., 2000] Russell, E. L., Chiang, L. H., and Braatz, R. D. (2000). *Data-Driven Techniques for Fault Detection and Diagnosis in Chemical Processes*. Springer.

[Silander and Myllymäki, 2006] Silander, T. and Myllymäki, P. (2006). A Simple Approach for Finding the Globally Optimal Bayesian Network Structure. In *proceedings of UAI*.

Part II

Probability Theory in Diagnosis

3

Bayesian Probability Theory

Probability is nothing but common sense reduced to calculation.

Laplace, 1812

In automotive dignosis, there are several sources of uncertainty: noise, model errors, lack of training data, etc. In this thesis we use probability theory to handle these uncertainties, and to determine faults that are possibly present in the monitored system. Rules for manipulating and updating probabilities are described for example in [Blom, 1994, Durrett, 2004, Casella and Berger, 2001]. However, one problem that remains when using probabilities to infer about the real world is to assign numbers to the probabilities. To do this, it is necessary to understand the word "probability". In this chapter, we briefly discuss different interpretations of probability and, in particular the interpretation of probability used in this thesis. This chapter is a shorter version of Appendix A.

3.1 Dealing With Uncertainty

Human life is to a great deal a life lived under uncertainty. Every day we make decisions under uncertainty, both in professional life and in private. For example: will the stock market raise or fall today? My car does not start, which part has caused the failure? Should I bring an umbrella tonight? How much should I bet on my favorite soccer team in the next game? Should I fold in the poker game? What conclusions can be drawn from the laboratory experiment? There is no upper limit on the number of such situations.

The situations listed above are very different in their nature. Sometimes the probability calculation relies on data, as in laboratory experiments. In other cases the probability calculations are based on known facts, for example, the number of spades in a deck of card is well known and thus the probability of drawing a spade can be computed. In yet other cases, it seems like probabilities are more or less based on personal feelings, for example in sports betting.

In each situation, the human brain deals with uncertainty. It considers the available information, for example: yesterday's stock market trend or the observation that the headlights of my car does not light. The brain weighs factors speaking fore and against an event, and makes decisions (which may be more or less clever).

In the problem considered in this thesis, diagnosis of automotive vehicles, we deal with uncertainty in a formal way. Given observations of different kinds from a system, the aim is to construct an algorithm that, just like the human brain, considers the available information and evaluates the probabilities that different faults are present. The available information can for example comprise data, different kinds of models with unknown model errors, drawings, and functionality specification documents. To be able to transform these fundamentally different types of information and construct the diagnosis algorithm that computes probabilities for faults, one might ask oneself questions as: What is this "uncertainty"? What is "probability"? What does the "probability that it will rain tonight" mean? Is it unique? Can we put a number on it?

In reference literature on probability theory, for example [Blom, 1994, Durrett, 2004, Casella and Berger, 2001, O'Hagan and Forster, 2004], formulas and tools for manipulating probabilities are presented, as in the following toy example.

Example 3.1.1 (Was it the Sprinkler?).
Sanna wakes up a morning and wants to know whether it has rained during the night. She knows that the prior probability for rain is $p(rain) = 0.3$. Moreover, she knows that, if it has rained, the lawn will be wet, i.e. that $p(wet\ lawn|rain) = 1$. She also knows that, if there is no rain, there is a sprinkler that cause the lawn to be wet with probability $p(wet\ lawn|no\ rain) = 0.2$.
After waking up, Sanna notices that the lawn is wet. She can then compute the probability that it has rained by using Bayes' rule and marginalization [Blom, 1994] as follows:

$$
\begin{aligned}
p(rain|wet\ lawn) &= \frac{p(wet\ lawn|rain)p(rain)}{p(wet\ lawn)} = \\
&= \frac{p(wet\ lawn|rain)p(rain)}{p(wet\ lawn|rain)p(rain) + p(wet\ lawn|no\ rain)p(no\ rain)} = \\
&= \frac{1 \cdot 0.3}{1 \cdot 0.3 + 0.2 \cdot 0.7} = 0.68\ldots
\end{aligned}
$$

These computations are perfectly fine as long as the numbers, such as "the probability for rain is 0.3", are known. In the example above, the numbers where simply stated, but how are they found? To assign numbers in the probability distributions to use in computations, it is necessary to know what "probability" means.

3.2 Interpretations of Probability

The discussion about the definition of the word "probability" has been going on for more than 200 years [Hacking, 1976]. Depending on the background of the researchers, there where several different interpretations during these years. Among the different interpretations of probability, there are two main paths [Hacking, 1976, O'Hagan and Forster, 2004, Jaynes, 2001]: the idea of probability as a *frequency* in an ensemble, often called the *frequentist* view or *frequency-type*, on the one hand; and the idea of probability as the *degree of belief* in a proposition, often referred to as the *Bayesian* view or *belief-type*, on the other hand. In frequentist view, probability is defined by the relative frequency of an event, and is a property of the object. Consider for example the statement:

> *This coin is biased towards heads. The probability of getting heads is about 0.6.*

This statement expresses probability in the frequency-type meaning, and is true depending on "how the world is". This statement can (at least hypothetically) be tested by tossing the coin (infinitely) many times. If the relative frequency for heads is 0.6, the statement is true, if the relative frequency for heads is something else, the statement is false. In the Bayesian view, probability is the degree of belief, given some evidence. Consider now this sentence about the same coin:

> *Taking all the evidence into consideration, the probability of getting a head in the next roll is about 0.6.*

This statement is true depending on how well evidence supports the particular probability assignment. The probability is subjective in the sense that it depends on the evidence. This statement can be true, depending on the evidence, even if the relative frequency turns out to be something else than 0.6.

These two views, the frequency-type and the belief-type, are different in a philosophical sense, and a natural question is why the same word, "probability", is used for both of them. Hacking [Hacking, 1976] gives one explanation: in

daily life, we (humans) switch back and forth between the two perspectives. Consider the following example.

Example 3.2.2 (Switching Between Frequency and Belief).
A truck of model R arrives to a mechanic at a workshop. The mechanic knows that among all model R trucks, one out of ten of the trucks that arrives to the workshop has fault F present. The mechanic concludes that choosing a random model R truck of those that has been (or are) at the workshop, the probability that fault F is found is 0.1. This probability is of frequency-type.
Consider now the particular truck that just arrived to the workshop. What is the probability that *this* truck is has fault F? The truck *is* either faulty of fault free, so there is no randomness, but still the mechanic would (probably) say that the probability is 0.1. He reasons as follwos. Out of all model R trucks that has visited the workshop, fault F was present in 1 out of 10. This truck is a model R and has arrived to the workshop. Taking those three pieces of information into account, the probability that this particular truck has fault F is 0.1.

The two interpretations of probability, as well as methods for assigning probabilities is further discussed in Appendix A. Instead, we now concentrate on the interpretation of probability used in this thesis.

3.3 The Interpretation of Probability Used in the Thesis

In this thesis, as in Example 3.2.2, we consider a specific vehicle. The vehicle is either fault-free or faulty, but since we, in general, not have enough information about the vehicle to determine its fault status, we use probabilities.

Although not being dogmatic, we will in this thesis mainly take a Bayesian, or belief-type, view on probability. We let the probability be determined by the evidence, or background information, given. To denote this, if $\mathbf{i}$ denotes all information given, we write the probability for an event A as $p(A|\mathbf{i})$. We let the probability be defined by is given behind the |-sign, i.e. by the evidence of *background information*. In this interpretation, the probability is subjective in the sense that different evidence give different probabilities. On the other hand, the probability is objective in the sense that we assume that it is *uniquely determined* about what is given behind the |-sign. This implies that we, to be formal, require enough information behind the |-sign to uniquely determine the probability. For example, if D denotes the number of eyes coming up when rolling a dice, the probability for getting six eyes in a certain trial is written

$$p(D = 6|S) = \frac{1}{6},$$

where

$$S = \textit{The dice is unbiased. The dice has six sided. We apply principle of indifference, that says that if there are } n \textit{ possible events and there is no reason for favoring any of the events over the others, each event should be assigned probability } 1/n.$$

The example above show a quite lengthy and intricate way of writing something that is implicitly understood. Furthermore, in many situations, it is uninteresting and/or extremely complicated to explicitly state every piece of information that is behind the $|$-sign. Therefore, we often simply denote this knowledge "background knowledge" (background information) and write $\mathbf{i}$. When the background knowledge is clear from the context we sometimes omit $\mathbf{i}$ as well.

We have, in this thesis, adopted the Bayesian interpretation of probability since it is appealing and natural for the reasoning in the problems related to diagnosis that we are faced to, or, as O'Hagan [O'Hagan and Forster, 2004] expresses it: "the Bayesian interpretation is fundamentally sound, very flexible, produce clear and direct inferences, and make use of all information"[1].

However, we are not dogmatic, and there are cases where the frequentist view is similar or equal. Technically, the rules of probabilities and the computations are the same in both interpretations of probabilities [Hacking, 1976]. This means that the methods presented in this thesis are valid and make sense regardless of the probability interpretation of the user.

[1]In contrast to classical methods that have "philosophical flaws", limited range, indirect interpretation of the inference, and not utilize prior information [O'Hagan and Forster, 2004].

4

A Brief Survey of Probability Based Diagnosis

4.1 Model-Based Diagnosis

4.1.1 Diagnosis Methods

During the last two decades, fault diagnosis of technical systems has become a steadily increasing field of research. One important reason is the introduction of more complex and capable computers and electronic control units (ECU), that mitigates improved system functionality that make the systems more difficult to diagnose. At the same time, the better ECUs provide a platform for improved diagnosis algorithms.

There is a huge number of different methods for doing diagnosis. In its most general form, diagnosis is to, based on knowledge about the system, study observations from the system and then draw conclusions about the state of the system. Different diagnosis methods are based on different "knowledge about the system" and consider observations in different ways.

In *model-based diagnosis* (MBD), models of the system under diagnosis are used to describe the relations between observations and faults, see Figure 4.1. The model typically describes how possible faults affect the observations. During diagnosis, these relations are inverted and the observations are used to draw conclusions about which faults that are present. There is a wide variety in model-types that can be used in diagnosis, and in Figure 4.2 an overview is given. This is by no means the only way of characterizing model-based diagnosis methods, and it is not complete, but gives an idea of some model-types that

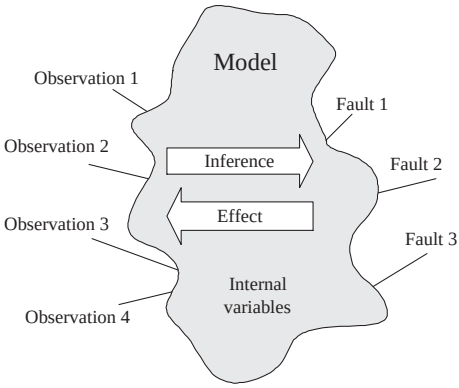


Figure 4.1: A diagnosis model describing how faults affect observations of the system. During diagnosis, observations are made ad the inverted relations are used to make inference about which faults that may be present.

appears in the literature. In the remainder of this section we present four types of models and a selection of works based on each of the model type.

4.1.2 Logical Models

Among the first modern methods for MBD we find Reiter's method based on first order logic [Reiter, 1992]. The system under diagnosis is described by using logical statements. In Reiter's method, the diagnoses are assignments of component states to all components in the system that are consistent with the observations made of the system. During the same time period as Reiter's method was developed, the General Diagnostic Engine (GDE) [de Kleer, 1992] and its descendant Sherlock [de Kleer and Williams, 1992] based on similar ideas where presented.

4.1.3 Black Box Models

Black box models, or *data driven models*, are learned from training data, and can for example be various classification methods [Duda et al., 2001, Devroye et al., 1996, Bishop, 2005, Russell et al., 2000, Chiang et al., 2001, Sorsa et al., 1991], among which we find for example Support Vector Machines (SVM) [Lee et al., 2007, Ge et al., 2004, Saunders et al., 2000], methods for Case Based Reasoning (CBR) [Bregon et al., 2007], and Bayesian networks learned from data [Verron et al., 2007, Pernestål et al., 2008]. Since data driven models are learned from data, they require no explicit knowledge about the process

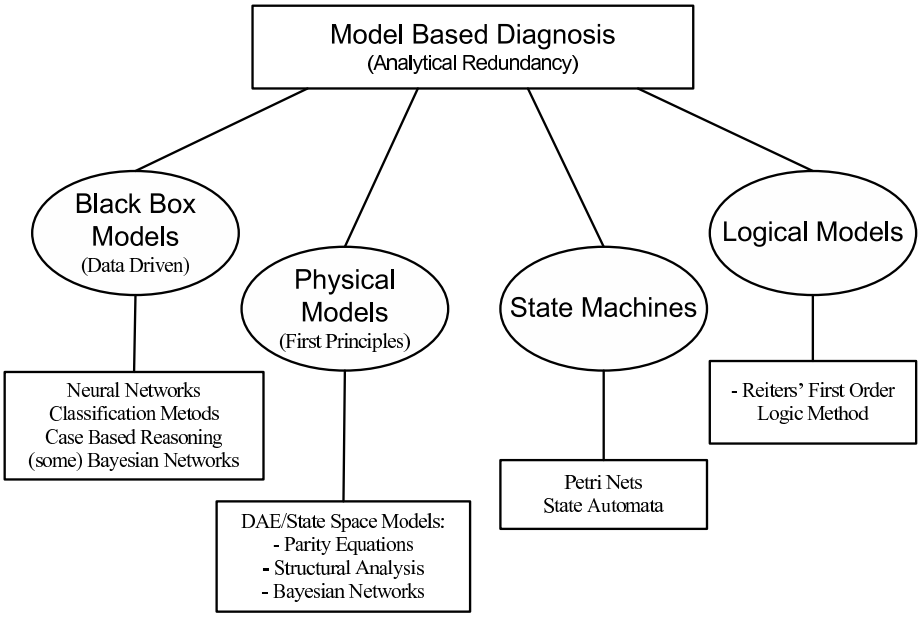


Figure 4.2: An overview of model based diagnosis methods.

under diagnosis. The main drawback with the data driven models in diagnosis is that they, in their general form, require data from all fault cases that are to be diagnosed – a situation that is rarely fulfilled in fault diagnosis applications since faults are rare.

4.1.4 Physical Models

Examples of physical model types are for example state space models and Differential Algebraic Equations (DAE). Physical models are used in diagnosis in several ways [Blanke et al., 2003, Patton et al., 2000, Isermann and Ballé, 2007, Isermann, 2006, Cordier et al., 2004, Staroswiecki and Comtet-Varga, 2001]. Among the diagnosis methods based on physical models we find for example parity space [Basseville and Nikiforov, 1993, Gertler, 1998, Zhang et al., 2006], structural analysis [Krysander, 2006], structural hypothesis testing [Nyberg, 2000], Bayesian network methods learned from physical principles [Roychoudhury et al., 2006, Schwall, 2005], and qualitative models [Daigle et al., 2007, Mosterman and Biswas, 1999]. In diagnosis using physical models, data is sometimes needed to tune the model, but the diagnosis result depend to a larger extent on the accuracy of the model than on the data. For automotive systems, the operation conditions and surroundings are continuously changing and it is typically difficult to build a model that is sufficiently accurate in all

operating conditions.

4.1.5 Discrete Event Systems

One large branch of diagnosis concerns diagnosis of Discrete Event Systems (DES), see for example the workshop series DCDS [Dotoli and Larizza, 2009]. When considering DES, the system is modeled by a set of states and transitions between these states [Kurien and Nayak, 2000]. Some states represent that the system is faulty. Diagnosis then becomes the task of tracking the sequence of states that system has been in, given observations from the system. Two commonly used model approaches are Petri nets [Murata, 1989, Aghasaryan et al., 1998] and state automatons [Lunze and Supavatanakul, 2002, Supavatanakul et al., 2006].

4.2 Probabilistic Methods for Diagnosis

4.2.1 An Example: the Car Start Problem

In this thesis, we apply *probabilistic methods* for diagnosis. Basically, this means that we compute probabilities for faults. The idea of using probabilistic methods for diagnosis is not new. In fact, diagnosis is one of the most common applications in introductory courses on probability theory. One example is the "Car start problem" [Jensen and Nielsen, 2007], where the task is to determine why a car does not start. A simple version of the car start problem is shown in Figure 4.3, where variables are shown as circles and dependencies between the variables are given by directed edges, point in the direction of causal influence. For example, the amount of fuel (*Fuel*?) and whether the starter rolls (*Starter Roll*?) have causal impact on the whether the car starts (*Car Start*?), and the state *Fuel*?. So, if probabilistic diagnosis problems can be solved in the basic course on probability, what is the problem? In the example above, the model, i.e. the causal dependencies between variables, is assumed to be known. This is typically not the case in real applications. Furthermore, dependencies need to be quantified. Finally, we need methods for inference, for example, to determine the probability that the fuel tank is empty, given that the car does not start and that the battery is fully charged. These three tasks are often challenging. In next section, we give a more precise formulation of the challenges in probabilistic diagnosis.

4.2.2 What is Probabilistic Diagnosis?

As stated in Chapter 1, the aim is to compute the probability distribution

$$p(\textit{system status}|\textit{all available information}), \tag{4.1}$$

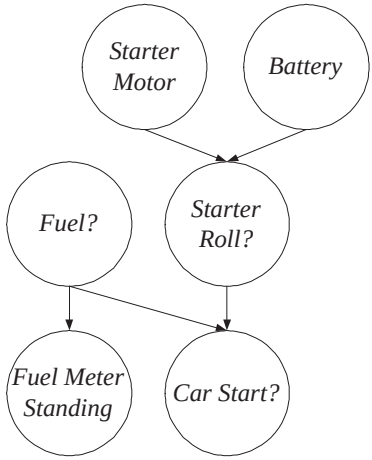


Figure 4.3: A basic example of diagnosis: the car start problem. The probability that the car starts is dependent on the fuel tank level (*Fuel*?), the battery status (*Battery*), and the status of the starter motor (*Starter Motor*).

where "all available information" may include current observations, training data, and other kinds of knowledge about the system. This distribution (4.1) is sometimes referred to as the *belief state*.

The knowledge about the system is often represented with some kind of model, for example one of those described in Section 4.1. Regardless of which kind of model that is used, it is often impossible determine exactly which faults that are present in the system. Reasons may for example be that the number of observation points is limited, that there are noise and model errors present, or the unknown and changing operating environment. These factors cause us to reason under uncertainty.

We divided the probabilistic diagnosis problem into two subproblems:

1. **Learning.** To construct, or learn, an adequate model of the system under diagnosis, including dependency structures and strength of dependencies.

2. **Inference.** To use the model to make inference, and compute the probability distribution for the system state, or for faults.

Depending on the model type used, these two steps will be more or less difficult.

4.3 Methods For Probabilistic Diagnosis

There are numerous methods for probabilistic diagnosis in the literature, based on different kinds of models. In this section, we review methods based on

three model-types that are most closely related to the methods presented in the appended papers in Part III. We discuss the kind of dependency relations and uncertainties that are modeled within each model type. We also consider the complexity of the two steps Learning and Inference, and summarize advantages and drawbacks.

4.3.1 Dynamic Physical Models

Model Type. Dynamic physical systems, such as combustion engines, automotive robots, chemical plants and many others, are often described by a state space model or by differential algebraic equations (DAEs) [Wahlström, 2009, Verma et al., 2004, Patton et al., 2000]. In a probabilistic setting, a discrete time state space model can for example written as

$$\begin{aligned} z_t &\sim p(z_t|z_{0:t-1}, x_{0:t-1}, y_{1:t-1}, u_{1:t-1}) \\ x_t &= f(z_t, x_{t-1}, w_t, u_t),\ w_t \sim p(w_t) \\ y_t &= g(z_t, x_t, v_t, u_t),\ v_t \sim p(v_t) \end{aligned}$$

where y_t are sensor readings, u_t known control signals, x_t continuous internal states, z_t discrete internal states, w_t and v_t are process and measurement noise respectively. In this model, y_t and u_t comprise the observations, and the faults states is a (subset) of z_t. All variables may be scalar or vector-valued.

Learning. Learning a dynamic physical model consists in determining the functions f and g, and distribution of the internal state z_t, and the distributions $p(w_t)$ and $p(v_t)$ of the noise w_t and v_t. The functions f and g are often equations representing the physical behavior of the system, and known by domain experts. The distribution $p(z_t|z_{0:t-1}, x_{0;t-1}, y_{1:t-1}, u_{1:t-1})$ describes transitions between discrete states in the system. The discrete variable z_t represents faults, and the probability for transitions is often assumed to be known. The distributions $p(w_t)$ and $p(v_t)$ are generally assumed to be known, and often considered to be Gaussian.

Inference. With this type of model, the belief state (4.1) that we search is the probability $p(z_t|y_{1:t-1}, u_{1:t-1})$. If the functions f and g are linear (or linearized), and v_t and w_t are (assumed to be) Gaussian the *Kalman Filter* can be used to determine the belief state, see for example [Gustafsson, 2001].

A more general approach, that applies to non-linear f and g, and non-Gaussian v_t and w_t, is the *Particle Filter* [Doucet et al., 2001], where the relevant distributions are approximated using a swarm of "particles", or realizations of $p(x_t|x_{0:t-1}, y_{1:t-1}, z_{0:t-1})$ and $p(y_t|x_{0:t}, y_{0:t-1}, z_{t-1})$. There are several

diagnosis applications based on particle filters, see for example [Freitas et al., 2003, Narasimhan et al., 2004, Verma et al., 2004, Koller and Lerner, 2000, Dearden and Clancy, 2002, Li and Kadirkamanathan, 2001].

Advantages. If the state space description is known and can be linearized, the Kalman Filter method is straight forward and computational efficient. State space models often exists for control, and these models can be reuse for diagnosis.

Drawbacks. The Kalman and Particle filters can often be used straightforwardly to detect abnormal behavior of the system. However, to isolate the particular fault that is present is often more challenging. Methods for diagnosis typically require multiple copies of the model and a bank of filters. This increases the computational burden. Furthermore, to isolate faults, models describing the effects of the faults on the process are needed.

4.3.2 Data-Driven Black Box Models

Model Type. Black Box models are learned from training data. The structure of the model does not aim to represent any physical relations between inputs and outputs. Examples of model types are given in Section 4.1.3. Sometimes, learning black box models is referred to as machine learning.

Learning. If no explicit information is known about the system under diagnosis, but there is a lot of *training data*, i.e. tuples of observations and corresponding fault statuses, from the system under diagnosis, there are methods for learning the black-box models presented in literature [Duda et al., 2001, Devroye et al., 1996, Bishop, 2005, Russell et al., 2000]. The methods are generally based on optimization of a performance measure by tuning parameters in the models. For a Bayesian approach, data can be used to learn a Bayesian network (BN) [Silander and Myllymäki, 2006], where the nodes in the BN represent observations and faults.

Inference. Depending on the type of black box model used, inference may be simple or complicated. However, in most of the methods, the learning part is the most time consuming, and designed to provide straight-forward inference. This is particularly true for regression methods and neural networks.

Advantages. No explicit knowledge about the process is needed.

Drawbacks. The main drawback with the data-driven black box probabilistic methods is that a large amount of data from all faults considered is needed. Often it is difficult to obtain data from the faulty cases, since faults are rare. The black box methods may also be difficult to interpret, and therefore they may also be difficult to verify. Even if there is knowledge of the process available, the existing methods for learning data driven probabilistic models can typically not integrate this information with the data.

4.3.3 Bayesian Networks

Model Type. A Bayesian network (BN) is a representation of a factorization of a joint distribution of a set of variables $X_1, \ldots, X_n$. A BN is a directed, acyclic graph, where nodes represent variables and edges between nodes represent dependency relations. To each node there is a conditional probability distribution (CPD) for the corresponding variable given its parents associated. Introductions to Bayesian networks are given for example in [Jensen and Nielsen, 2007] and [Russell and Norvig, 2003].

Learning. In literature several ways of learning BNs for diagnosis are presented. The most common are: to learn from data, see Section 4.3.2; to use BNs set up by experts as in [Lerner et al., 2000, Schwall, 2005]; or to systematically derive the BNs from sets of physical equations by using a bond graph [Roychoudhury et al., 2006]. Also, for a given dependency, the CPDs can be learned from data.

The structures of the BNs used for diagnosis in the literature are different. Some of the most common are: two-layer BNs, where the nodes are either observations (in terms of sensor signals, residuals or diagnostic tests) or components as in [Schwall, 2005, Verron et al., 2009], multilayer BNs including internal variables and capturing the structure of the system as in [Schwall and Gerdes, 2002], and dynamic Bayesian networks (DBN) capturing the dynamics of systems [Murphy, 2002, Roychoudhury et al., 2006].

Inference. When the BN is known, standard methods can be used for inference. The most common are variable elimination and join tree. For large BNs with many nodes and many dependencies the inference methods may become computationally intractable and approximation methods must be applied [Jensen and Nielsen, 2007]. Methods for learning DBNs are presented in [Murphy, 2002].

Advantages. BNs representing physical structures are usually easy to interpret and validate. Also, they can be easily updated with local changes if the system under diagnosis is changed [Russell and Norvig, 2003].

Drawbacks. In some systems there may be several unknown and hidden effects. These may be difficult to learn and model in the BN, but may be important for the diagnosis result [Pernestål et al., 2006]. Furthermore, even if dependency structures of BNs for diagnosis by experts, learning the numbers in the CPDs is often more difficult since standard methods for parameter learning, as for example in [Heckerman et al., 1995] require data from all faults to be detected.

References

[Aghasaryan et al., 1998] Aghasaryan, A., Fabre, E., Benveniste, A., and Jard, C. (1998). Fault Detection and Diagnosis in Distributed Systems: An Approach by Partially Stochastic Petri Nets. *Discrete Event Dynamic Systems*, 8:203 – 231.

[Basseville and Nikiforov, 1993] Basseville, M. and Nikiforov, I. V. (1993). *Detection of Abrupt Changes. Theory and Application*. Prentice Hall, New Jersey.

[Bishop, 2005] Bishop, C. M. (2005). *Neural Networks*. Oxford University Press.

[Blanke et al., 2003] Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., and Schröder, J. (2003). *Diagnosis and Fault Tolerant Control*. Springer, New York.

[Blom, 1994] Blom, G. (1994). *Sannolikhetsteori och statistik med tillämpningar*. Studentlitteratur.

[Bregon et al., 2007] Bregon, A., Pulido, B., Simon, A., Moro, Q. I., Prieto, O.-J., Rodriguez, J. J., and Alonso, C. (2007). Focusing Fault Localization in Model-based Diagnosis with Case-based Reasoning. In *Proceedings of the European Control Cinference*.

[Casella and Berger, 2001] Casella and Berger (2001). *Statistical Inference (2nd edition)*. Duxbury Press.

[Chiang et al., 2001] Chiang, L., Braatz, R. D., and Russell, E. L. (2001). *Fault Detection and Diagnosis in Industrial Systems*. Springer.

[Cordier et al., 2004] Cordier, M.-O., Dague, P., Levy, F., Montmain, J., Staroswiecki, M., and Trave-Massuyes, L. (2004). Conflicts Versus Analytical Redundancy Relations: a Comparative Analysis of the Model Based Diagnosis Approach from the Artificial Intelligence and Automatic Control Perspectives. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(5):2163–2177.

[Daigle et al., 2007] Daigle, M., Koutsoukos, X. D., and Biswas, G. (2007). A qualitative approach to multiple fault isolation in continuous systems. In *AAAI*, pages 293–298.

[de Kleer, 1992] de Kleer, J. (1992). Focusing on Probable Diagnosis. *Readings in model-based diagnosis*, pages 131–137.

[de Kleer and Williams, 1992] de Kleer, J. and Williams, B. C. (1992). Diagnosis with Behavioral Modes. In *Readings in Model-based Diagnosis*, pages 124–130, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Dearden and Clancy, 2002] Dearden, R. and Clancy, D. (2002). Particle Filters for Real-Time Fault Detection in Planetary Rovers. In *Proceedings of 13th International Workshop on Principles of Diagnosis (DX 02)*, pages 1–6, Semmering, Austria.

[Devroye et al., 1996] Devroye, L., Györfi, L., and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer, New York.

[Dotoli and Larizza, 2009] Dotoli, M. and Larizza, P. (2009). *Proceedings of 2nd IFAC Workshop on Dependable Control of Discrete Systems*.

[Doucet et al., 2001] Doucet, A., Freitas, N. D., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer.

[Duda et al., 2001] Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley, New York.

[Durrett, 2004] Durrett, R. (2004). *Probability: Theory and Examples*. Duxbury Press.

[Freitas et al., 2003] Freitas, N. D., Dearden, R., Hutter, F., Morales-menendez, R., Mutch, J., and Poole, D. (2003). Diagnosis by a waiter and a mars explorer. In *In Invited paper for Proceedings of the IEEE, special*, page 2004.

[Ge et al., 2004] Ge, M., Du, R., Zhang, G., and Xu, Y. (2004). Fault diagnosis using support vector machine with an application in sheet metal stamping operations. *Mechanical Systems and Signal Processing*, 18(1):143 – 159.

[Gertler, 1998] Gertler, J. J. (1998). *Fault Detection and Diagnosis in Engineering Systems*. Marcel Decker, New York.

[Gustafsson, 2001] Gustafsson, F. (2001). *Adaptive Filtering and Change Detection*. Wiley.

[Hacking, 1976] Hacking, I. (1976). *The Logic of Statistical Inference*. Cambridge University Press.

[Heckerman et al., 1995] Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20(3):197–243.

[Isermann, 2006] Isermann, R. (2006). *Fault-Diagnosis Systems*. Springer, Germany.

[Isermann and Ballé, 2007] Isermann, R. and Ballé, P. (2007). Trends in the Application of Model-Based Fault Detection and Diagnosis of Technical Processes. *Readings in model-based diagnosis*, 37(3):348–361.

[Jaynes, 2001] Jaynes, E. T. (2001). *Probability Theory - the Logic of Science*. Camebridge University Press, Cambridge.

[Jensen and Nielsen, 2007] Jensen, F. V. and Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs*. Springer.

[Koller and Lerner, 2000] Koller, D. and Lerner, U. (2000). Sampling in factored dynamic systems. In *Sequential Monte Carlo Methods in Practice*, pages 445–464. Springer-Verlag.

[Krysander, 2006] Krysander, M. (2006). *Design and Analysis of Diagnosis Systems Using Structural Methods*. PhD thesis, Linköping University, Linköping, Sweden.

[Kurien and Nayak, 2000] Kurien, J. and Nayak, P. P. (2000). Back to the Future for Consistency-based Trajectory Tracking. In *Proceedings of AAAI*.

[Lee et al., 2007] Lee, G., Bahri, P., Shastri, S., and Zaknich, A. (2007). A Multi-Category Decision Support System Framework for the Tennessee Eastman Problem. In *Proceedings of the European Control Conference (ECC 07)*.

[Lerner et al., 2000] Lerner, U., Parr, R., Koller, D., and Biswas, G. (2000). Bayesian Fault Detection and Diagnosis in Dynamic Systems. In *AAAI/IAAI*, pages 531–537.

[Li and Kadirkamanathan, 2001] Li, P. and Kadirkamanathan, V. (2001). Particle Filtering Based Likelihood Ratio Approach to Fault Diagnosis in Nonlinear Stochastic Systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 31(3):337–343.

[Lunze and Supavatanakul, 2002] Lunze, J. and Supavatanakul, P. (2002). Diagnosis of discrete event system described by timed automata. In *Proceedings of the 15th IFAC World Congress*.

[Mosterman and Biswas, 1999] Mosterman, P. J. and Biswas, G. (1999). Diagnosis of continuous valued systems in transient operating regions. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 29(6):554–565.

[Murata, 1989] Murata, T. (1989). Petri nets: Properties, Analysis and Applications (Invited Paper). In *Proceedings of the IEEE*, pages 541–58.

[Murphy, 2002] Murphy, K. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, UC Berkeley, USA.

[Narasimhan et al., 2004] Narasimhan, S., Dearden, R., and Benazera, E. (2004). Combining particle filters and consistency-based approaches for monitoring and diagnosis of stochastic systems hybrid. In *Proceedings of 15th International Workshop on Principles of Diagnosis (DX 04)*.

[Nyberg, 2000] Nyberg, M. (2000). Model Based Fault Diagnosis Using Structured Hypothesis Tests. In *Fault Detection, Supervision and Safety for Technical Processes. IFAC*, Budapest, Hungary.

[O'Hagan and Forster, 2004] O'Hagan, A. and Forster, J. (2004). *Kendall's Advanced Theory of Statistics*. Arnold, London.

[Patton et al., 2000] Patton, R. J., Frank, P. M., and Clark, R. N. (2000). *Issues of Fault Diagnosis for Dynamic Systems*. Springer, New York.

[Pernestål et al., 2006] Pernestål, A., Nyberg, M., and Wahlberg, B. (2006). A Bayesian Approach to Fault Isolation with Application to Diesel Engine Diagnosis. In *Proceedings of 17th International Workshop on Principles of Diagnosis (DX 06)*, pages 211–218.

[Pernestål et al., 2008] Pernestål, A., Wettig, H., Silander, T., Nyberg, M., and Myllymäki, P. (2008). A bayesian approach to learning in fault isolation. In *Proceedings of the 19th International Workshop on Principles of Diagnosis*.

[Reiter, 1992] Reiter, R. (1992). A Theory of Diagnosis From First Principles. In *Readings in Model-based Diagnosis*, pages 29–48, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Roychoudhury et al., 2006] Roychoudhury, I., Biswas, G., and Koutsoukos, X. (2006). A Bayesian Approach to Efficient Diagnosis of Incipient Faults. In *17th International Workshop on Principles of Diagnosis (DX 06)*, pages 243–250.

[Russell et al., 2000] Russell, E. L., Chiang, L. H., and Braatz, R. D. (2000). *Data-Driven Techniques for Fault Detection and Diagnosis in Chemical Processes*. Springer.

[Russell and Norvig, 2003] Russell, S. and Norvig, P. (2003). *Artificial Intelligence. A Modern Approach*. Prentice Hall.

[Saunders et al., 2000] Saunders, C., Gammerman, A., Brown, H., and Donald, G. (2000). Application of Support Vector Machines to Fault Diagnosis and Automated Repair. In *Proceedings of 11th International Workshop on Principles of Diagnosis (DX 00)*.

[Schwall, 2005] Schwall, M. (2005). *Dynamic Integration of Probabilistic Information for Diagnostics and Decisions*. PhD thesis, Stanford University, Stanford University.

[Schwall and Gerdes, 2002] Schwall, M. and Gerdes, C. (2002). A probabilistic Approach to Residual Processing for Vehicle Fault Detection. In *Proceedings of the 2002 ACC*, pages 2552–2557.

[Silander and Myllymäki, 2006] Silander, T. and Myllymäki, P. (2006). A Simple Approach for Finding the Globally Optimal Bayesian Network Structure. In *proceedings of UAI*.

[Sorsa et al., 1991] Sorsa, T., Koivo, H. N., Member, S., and Koivisto, H. (1991). Neural networks in process fault diagnosis. *IEEE Transactions on Systems, Man and Cybernetics*, 21:815–825.

[Staroswiecki and Comtet-Varga, 2001] Staroswiecki, M. and Comtet-Varga, G. (2001). Analytical Redundancy Relations for Fault Detection and Isolation in Algebraic Dynamic Systems. *Automatica*, 27:687–699.

[Supavatanakul et al., 2006] Supavatanakul, P., Lunze, J., Puig, V., and Quevedo, J. (2006). Diagnosis of timed automata: Theory and application to the damadics actuator benchmark problem. *Statistics & Probability Letters*, 14:609–619.

[Verma et al., 2004] Verma, V., Gordon, G., Simmons, R., and Thrun, S. (2004). Particle filters for rover fault diagnosis. *IEEE Robotics and Automation Magazine*.

[Verron et al., 2007] Verron, S., Tiplica, T., and Kobi, A. (2007). Fault Diagnosis of Industrial Systems with Bayesian Networks and Mutual Information. In *Proceedings of the European Control Conference (ECC 07)*, pages 2304–2311.

[Verron et al., 2009] Verron, S., Weber, P., Theilliol, D., Tiplica, T., Kobi, A., and Aubrun, C. (2009). Decision with Bayesian Network in the Concurrent

Faults Event. In *Proceedings of 7th IFAC Symposium on Fault Detection, Supervision and Safety of technical processes (SAFEPROCESS 2009)*.

[Wahlström, 2009] Wahlström, J. (2009). *Control of EGR and VGT for Emission Control and Pumping Work Minimization in Diesel Engines*. PhD thesis, Linköping University, Linköping, Sweden.

[Zhang et al., 2006] Zhang, P., Ye, H., Ding, S., Wang, G., and Zhou, D. (2006). On the relationship between parity space and H2 approaches to fault detection. *System and Control Letters*.

Part III

Papers

Paper 1

Bayesian Fault Diagnosis for Automotive Engines by Combining Data and Process Knowledge[1]

Anna Pernestål and Mattias Nyberg

Division of Vehicular Systems, Department of Electrical Engineering,
Linköping University,
Sweden.

Abstract

We consider fault diagnosis of complex systems, motivated by the problem of fault diagnosis of an automotive diesel engine. Previous fault diagnosis algorithms are typically based either on process knowledge, for example a Fault Signature Matrix (FSM), or on training data. Both these methods have their advantages and drawbacks. The main contribution in the present work is that we show how to integrate process knowledge and training data to improve fault diagnosis for automotive processes. We carefully investigate the characteristics of our motivating application, and we derive a new method for fault diagnosis based Bayesian inference. To illustrate the new fault diagnosis method we have applied it to the diagnosis of the gas flow of an automotive engine using data from real driving situations. It is shown that diagnosis performance is improved compared to previous methods using solely data or process knowledge. Finally we study the relation between the new method and previous state of the art methods for fault diagnosis.

[1]This paper has been submitted to IEEE Transactions on Systems, Man, and Cybernetics Part A. It is based on the papers [Pernestål and Nyberg, 2007b].

1 Introduction

Fault diagnosis is the task of detecting and localizing faults currently present in a process, given a set of observations from the process. The observations can for example be sensor readings, residuals, or different kinds of diagnostic tests.

When faults occur, appropriate counter actions should be performed to avoid accidents, maintain usability, and minimize repair costs. The aim of diagnosis is to provide enough information so that this can be done in an optimal way. Here we present a probability based method for fault diagnosis, where the probabilities for different faults are computed given all information available. The probabilities can then be combined with decision theoretic methods to determine the best action given the current situation.

Many approaches to fault diagnosis have been proposed in literature. Typically, they either rely on a model of the process, so called model based diagnosis (MBD), as e.g. in [Gertler, 1998, de Kleer and Williams, 1992, Reiter, 1992, Schwall and Gerdes, 2002, Narasimhan and Biswas, 2007], or they are data driven e.g. using statistical or pattern recognition methods as in [Fouladirad and Nikiforov, 2005, Verron et al., 2007, Lee et al., 2007]. All these methods have their advantages and drawbacks. The MBD methods utilize a model of the relations between faults and observations, e.g. differential equations [Gertler, 1998], logical models [de Kleer and Williams, 1992, Reiter, 1992], or probabilistic models [Schwall and Gerdes, 2002, Narasimhan and Biswas, 2007], and they perform well on processes where detailed and accurate models exist. However, this is generally not the case for real, complex processes. The data driven fault diagnosis methods, on the other hand, require no model of the process. Instead, they use a lot of training data from all possible faults. The main drawback with data driven method is that often there is only a limited amount of data available since faults occur rarely or are sometimes even unknown.

Our motivating application is diagnosis of an automotive diesel engine. Our aim is not to solve the diagnosis task for this specific application, but rather to use the application as inspiration to develop a new, general method for fault diagnosis, suitable for different automotive applications. When studying the diesel engine application, we have realized that process knowledge gives models with performance insufficient for MBD methods. Furthermore, the amount of data, especially from situations with faults present, is limited. However, an important fact is that there are both process knowledge *and* training data available. Inspired by this fact, our approach is to integrate process knowledge and data, which each are insufficient, into one combined diagnosis method with improved performance.

In automotive processes, the aim of diagnosis is to provide information to choose the best possible action to avoid accidents and maintain usability. We therefore focus on computing the probabilities for different faults. The probabilities can then be combined with decision theoretic methods as the ones described

for example in [Heckerman et al., 1995a, Warnquist et al., 2009, Langseth and Jensen, 2002] to find the optimal counter action.

This new, combined method for diagnosis is the main contribution in the paper. In order to develop the method we first carefully investigate the characteristics of automotive diagnosis problems and formulate the diagnosis problem in terms of probabilities. In particular, we discuss underlying assumptions explicitly. We focus on process knowledge expressed as a Fault Signature Matrix (FSM), and give an interpretation of it in the probabilistic framework. To illustrate the power of the new method we apply it to diesel engine diagnosis, using training data from a real driving situation.

There are previous works on combining data and process knowledge for diagnosis, see e.g. [Bregon et al., 2007, Alonso-González et al., 2008, Becraft et al., 1991]. The two main differences are that these previous works typically requires more training data, and, most importantly, we compute the probabilities for faults while they provide classification.

The paper is outlined as follows. First, we explain the characteristics of our motivating application and its requirements on the diagnosis method, and discuss related work in Section 2. In Section 3 is the problem formulated formally, and in in Section 4 is the knowledge available for fault diagnosis is studied in detail. The computations of the probabilities for different faults given training data only is presented in Section 5, and in Section 6 the computations are extended to also take process knowledge into account. In Section 7 the method is illustrated on the problem of fault diagnosis on the automotive diesel engine, using training and evaluation data from real driving situations. In Section 8 we discuss practical design choices. Finally, we give a thorough investigation of the relations to previous model-based fault diagnosis methods in Section 9 before we conclude the paper in Section 10.

2 The Automotive Diagnosis Problem

One of our main objectives with the work presented in this paper, besides that it should be general and theoretically sound, is that it should be applicable on real world automotive processes, and tackle the challenges associated with such applications. We let the work be motivated by the diagnosis of a heavy truck diesel engine. Here we provide the background of the application, and describe its specific characteristics and requirements in Section 2.1, before we summarize the requirements on the diagnosis method in Section 2.2, and present some previous work in Section 2.3. A concrete description of the process studied, the gas flow of a heavy truck diesel engine, is given in Section 7.

2.1 Motivating Application

Characteristics of the Automotive Processes

The processes we consider, automotive processes such as combustion engines, EGR[2] systems, SCR[3] catalysts, and particle filters are often large and complex. They are often difficult to model, both due to their complexity, but also since they consist of several different parts: there are chemical processes, thermodynamics, mechanics, mechatronics etc. Furthermore, there are often large vehicle-to-vehicle variations due to tolerances, aging and rebuilding of vehicles. Finally, automotive systems operate in continuously varying surroundings: the environment such as humidity and ambient pressure, the operation point, the driver's behavior. All this make detailed models and descriptions of the processes unreliable.

Fault Characteristics

Automotive diagnosis is of course important for detecting safety related faults and avoid accidents, but the main part of diagnosis in production engines is related to emission and performance monitoring. For example, a fault in the ambient temperature sensor may cause erroneous control of the after treatment system which in turn results in increased NOx emissions. Another example is a clogged exhaust gas pipe that in turn causes increased fuel consumption. In contrast to many safety related faults, the emission and performance related faults have a relatively slow and small impact on the process. On the other hand, while safety related faults must be detected and isolated quickly to avoid accidents, it is often enough to detect and isolate emission and performance related faults within a couple of hours.

The main challenge when diagnosing emission and performance related faults is rather their small impact on the system in combination with the difficulty of modeling the systems accurately.

On-Line Information

To control the automotive process it is equipped with sensors measuring e.g. temperatures, pressures, flows and actuator positions. These sensor readings can be used for diagnosis of the process. However, in many automotive applications there already exists *monitoring functions*, or monitors, that are designed to detect faults that appears. A monitor is a function of a set of measured signals, and is designed to change behavior (typically mean or variance) when faults appear. Monitors can for example be based on physical or logical models of subparts of the process, engineering skills and specific knowledge of the process,

[2]Exhaust Gas Recirculation
[3]Selective Catalytic Reduction

hardware redundancy, signal-in-range-checks, or state-machines. They may be delivered by subsystem suppliers, or by development engineers. The monitors may be binary, discrete or continuous, and many of them are results several man-years of research and development. Many of the monitors are black-box systems. Diagnostic tests and residuals are typical examples of monitors

The sensor readings could be used directly in the diagnosis. However, using monitors is generally more efficient since in these information important for diagnosis is extracted from the sensor readings. For example, for diagnosis it is often more interesting to know the relation between to sensor values rather than their actual values.

For the diesel engine studied in the current paper, there are several hundred of monitors available. It is important to notice that although a monitor is designed to detect a specific fault, it does not certainly do so due to noise and uncertainties. Furthermore, a monitor may be designed to detect a subset of faults, and several different monitors may be designed to detect the same fault. Thus, there is no one-to-one relation between faults and monitors. In order to choose the correct counter action it is necessary to collect the information from all monitors and compute the probability that different faults are present.

Training Data

Beside the monitoring functions and sensor readings, there is typically also some training data available. Training data consists of samples of monitor outputs together with information about the current status of the truck. There is often training data available from the fault free case, but data from fault situations is much more rare. The reason is that the diagnosis method should be implemented (and parameters in the methods set) in the vehicle control unit during the product development phase. At this stage only a few test vehicles are available, and training data can mainly be collected from the fault free case. To gain training data from faulty cases, faults may be implemented and data collected. However, some faults are dangerous or even impossible to implement. Furthermore, implementing many faults and combinations of faults is extremely time consuming and thus infeasible. Therefore the amount of training data available for the diagnosis method is typically limited, only available from some faults, and is experimental (not distributed according to the prior probability for the different modes). These restrictions on training data leads to that standard methods for learning (see e.g. [Heckerman et al., 1995b, Duda et al., 2001] can not be used since they generally requires more data and that data is observational (i.e. non-experimental).

Computational Limitations

On-board processors of heavy trucks have limited processor and storage capacities (128 kB internal memory, 1 MB external memory and 128 MHz processor are typical values), which limits both the number of computations that can be performed on-board as well as the amount of data that can be stored. Although both processor and storage capacities of on-board processors are increasing, low-capacity processors will always be more robust and cheaper, and motivates the development of low-complexity diagnosis algorithms [Cascio et al., 1999].

2.2 Requirements on the Solution

In both on- and off-board diagnosis of automotive processes, when an abnormal situation appears, the objective is to determine the best action to perform to return the process to an efficient and safe operation state. We believe that such a choice is best performed using decision theoretic algorithms, where the *probabilities* that different faults are present combined with a cost function. Another advantage with decision theory is that it makes it possible to compute the expected costs of the different faults using the monitoring functions available. Solving the decision theoretic problem is indeed interesting, but strongly application dependent and a research area in itself, see e.g. [Heckerman et al., 1995a, Warnquist et al., 2009, Langseth and Jensen, 2002] for some examples where the expected cost of repair is competed.

A fault or combination of faults is called a mode, and the focus in the current paper is to derive a generic method for computing the probabilities that different modes are present in an automotive process, given the current observations of the monitoring functions and taking into account restrictions that are common in automotive applications. We consider the probabilities for the different faults as the output from our fault diagnosis method, and a mode with non-zero probability is called a *diagnosis*. To make the probability computations as good as possible, we aim at using all information available. The information at hand is more carefully described in Section 4, but basically a mixture of engineering knowledge about constraints on the monitoring functions and training data.

We summarize the discussion above with listing the requirements on the fault diagnosis method for automotive processes.

- The probabilities for different faults should be computed, using all information available.
- Available information comprise training data, different kinds of monitoring functions, and possibly also sensor readings.
- The method should handle that the amount of training data available for learning may be limited.

- The method should handle that training data often is experimental.
- There is no detailed and accurate model of the whole system available.
- The computational burden should be kept small to meet specifications of ECU-processors.

2.3 Related Work

Before going into the details of the current method we first provide a short survey of previous work on automotive diagnosis and a brief comparison with previous work on combining data and knowledge and other closely related diagnosis methods.

Automotive Diagnosis

Automotive diagnosis is important and challenging, and is indeed previously studied in the literature. Several previous works are based on different kinds of models of the processes. In [Lee et al., 2005] model-based diagnosis of a spark ignition engine in a production vehicle is presented, in [Gertler et al., 1995] parity equations generated from nonlinear dynamic engine model are used to isolate actuator and sensor faults, in [Cascio et al., 1999] qualitative models are used, and in [Schwall and Gerdes, 2002] Bayesian networks. The diagnosis performance of these previous works is of course dependent on the quality of the model. These previous works typically study smaller subparts of the processes, where sufficiently good models exist. In the current application, we search a general method to operate on larger processes and where no sufficiently good model exist. Furthermore, with exception of the Bayes net based methods, they do not provide the probabilities for faults.

There are also some previous work on data driven diagnosis of automotive systems. For example in [Vemuri and Polycarpou, 1997] the limited accuracy of models is considered, and observational data from all modes is used to train a neural network for diagnosis. Such data does not exist in the current application.

These previous works on automotive diagnosis also points out several important application related features which we also have noted and tries to tackle. Examples are the processor limitations, vehicle aging, vehicle-to-vehicle differences [Cascio et al., 1999, Lee et al., 2005].

Combining Data and Knowledge

The approach of combining data and process knowledge for diagnosis is previously studied in e.g. [Bregon et al., 2007, Alonso-González et al., 2008]. In these works process knowledge is first used in terms of an FSM, and then the result is focused using classification techniques. Another example is [Becraft

Table 1: Sensors in the diesel engine

sensor	description
p_{em}	exhaust gas pressure
p_{im}	inlet manifold gas pressure
T_{im}	inlet manifold temperature
p_{amb}	ambient pressure
T_{amb}	ambient temperature
u_{EGR}	EGR valve position
w_{cmp}	flow through the compressor
n_{eng}	engine speed
n_{trb}	turbine speed

et al., 1991], where data based Neural Network methods are applied, and then expert knowledge is used refine the diagnosis. These previous works on combining data and process knowledge typically rely on the existence of data from all faults that are considered. Furthermore, they provide classification of rather than computation of the probability for the faults. In contrast, the Bayesian approach to fault diagnosis suggested here provides a solid ground for probability computations and a unified framework that can take advantage also of a limited amount of data.

Utilizing the Monitoring Functions

In many previous works on diagnosis, monitoring functions are thresholded and used to set up a binary FSM [Gertler, 1998, Korbicz et al., 2004]. However, if the FSM is extended by considering also the magnitude and direction of change, more information can be gained from the monitors. This is the aim in the current paper, as well as in [Pulido et al., 2005, Narasimhan and Biswas, 2007, Zhao et al., 2005]. In these previous works the direction and magnitude of change are to be specified in the design, while in the current work they are learned from data.

2.4 Example Application: The Diesel Engine

A schematic figure of the motivating application, the gas flow of the diesel engine is depicted in Figure 1. In total there are more than hundred possible faults that may appear on the engine. However, in order to make the results easier to overview we consider only faults in nine of the sensors in the engine, listed in Table 1, as well as previously unknown faults, i.e. faults with unknown effects.

We consider a set of ten monitors, that are representative and easily run in our test environment. The monitors are sensitive to different subsets of the

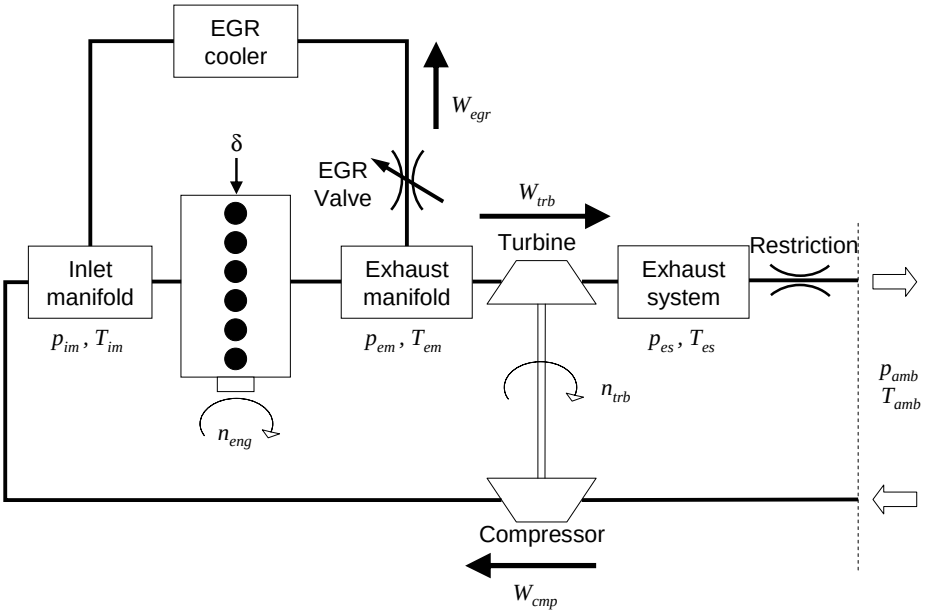


Figure 1: A schematic figure of the gas flow through a diesel engine with EGR

faults. In Figure 2, one of the monitors, r_4, is plotted. In the upper plot, a fault in w_{cmp} appears at $t = 700s$, and in the lower plot, a fault in T_{im} appears at $t = 620s$. As seen in Figure 2, the monitoring function output may differ from zero even in the fault free case (before $t = 620s$), i.e. there are model errors. These errors depend on unmodeled factors such as driver's behavior, operation point, and humidity. In our application, the monitors are based on continuous residuals, which means that they must be sampled and discretized.

To illustrate the nature of the observations, two of the sampled monitoring function outputs and the thresholds used for each of them are plotted in Figure 3 for six different faults. The observations corresponding to these two monitors are in the following referred to as $\mathbf{X}_3$ and $\mathbf{X}_4$. The observation $\mathbf{X}_4$ is the discretization of monitor r_4. Exact where to put the bin edges in the discretization, and how many bins to use is considered as a design parameter and discussed further in Section 8.

3 Problem Formulation

As stated in Section 2 the fault diagnosis task is to determine the probability distribution for the modes of the process, given the current observations and all other information at hand. In this section we introduce the necessary notation

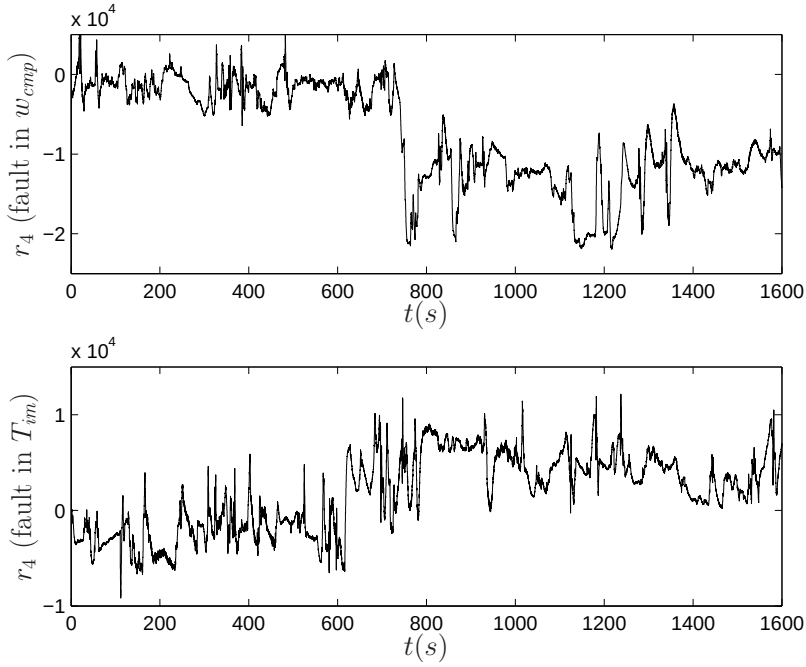


Figure 2: One of the monitoring functions, when faults in w_{cmp} (upper) and in T_{im} (bottom) appears.

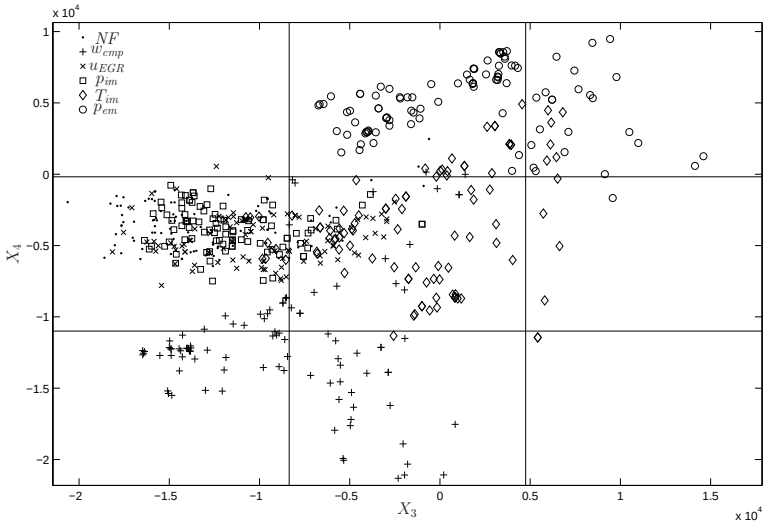


Figure 3: The observations X_3 and X_4 for six different modes, including the fault free mode.

and give a formal problem formulation.

The process under diagnosis is assumed to be in one of several predefined modes, which are characterized by one or several faults that are present, e.g. "leakage in pipe P_1" or "bias in sensor S_1 and leakage in pipe P_1". There are also the modes "no fault", where everything is functioning correctly, and "unknown fault".

3.1 Notation

At a certain instant j the system under diagnosis is described by two discrete variables: a scalar mode variable C^j, and an observation vector $\mathbf{X}^j = (X_1^j, \ldots X_R^j)$. Sometimes it may be more intuitive to characterize the mode variable as a vector where each element denotes the presence or absence of a certain fault. If a vector is used, the different modes can be enumerated, so without loss of generality we can assume that C^j is scalar.

The mode variable C^j can take the values $c_1, \ldots, c_L$. A value c_i is called a *mode*, and represents one of the predefined modes of the process. The modes are mutually exclusive.

Each element X_l^j in the observation vector $\mathbf{X}^j$ can take K_l different values and has the domain $\mathbb{X}_l = \{x_{l1}, \ldots, x_{lK_l}\}$. Consequently, the observation vector $\mathbf{X}^j$ has domain $\mathbb{X} = \mathbb{X}_1 \times \mathbb{X}_2 \times \ldots \times \mathbb{X}_R$. To denote an assignment of the

observation vector we write $\mathbf{X}^j = \mathbf{x}_k$, $k = 1, \ldots, K$, where $K = \prod_{l=1}^{R} K_l$. Each value $\mathbf{x}_k$ is a vector, and we write $\mathbf{x}_k = (\mathbf{x}_k[1], \ldots, \mathbf{x}_k[R])$ to denote the elements explicitly. With this notation $\mathbf{x}_k[i]$ is the value of X_i^j, when the value of $\mathbf{X}^j$ is $\mathbf{x}_k$.

For example, with $\mathbf{X}^j = (X_1^j, X_2^j)$, and $X_i^j \in \{x_{i1}, x_{i2}\}$, $i = 1, 2$ we have $K_i = 2$ and $K = 4$. Furthermore, if $\mathbf{x}_1 = (x_{11}, x_{12})$, then $\mathbf{x}_1[1] = x_{11}$ and $x_1[2] = x_{12}$.

Sometimes we will also use the notation $\mathbf{X}^j = \mathbf{x}^j$ and $C^j = c^j$ to stress that $\mathbf{x}^j$ and c^j are the values of the observation and the mode at sample j.

For discrete probability distributions we use the notation $p(Y = y|\mathbf{I})$, and for continuous probability density functions we use the notation $f(y|\mathbf{I})$. Here, $\mathbf{I}$ denotes the *background information.* In the current work we have adopted the view of probability as described for example in [Jaynes, 2001], where the probability is uniquely determined by the information given behind the "|"-sign. For the sake of brevity we will not write out the background information in the formulas, although it is conceptual important, and thus we write $p(Y = y)$ or $f(y)$.

3.2 Formal Problem Formulation

The objective is to determine the probabilities for different modes, given the information at hand. The information at hand includes training data, denoted $\mathcal{D}$, as well as other knowledge about the process to be diagnosed, denoted $\mathbf{i}_\mathcal{R}$. As stressed in Section 1, both training data and process knowledge are important for fault diagnosis.

Assume that r consecutive observation vectors are collected and that the same fault is present during the collection of these samples. We can now state the fault diagnosis problem formally as to compute

$$p(C^{J_r} = c_i|\mathbf{X}^{J_1} = \mathbf{x}_{k_1}, \mathbf{X}^{J_2} = \mathbf{x}_{k_2}, \ldots, \mathbf{X}^{J_r} = \mathbf{x}_{k_r}, \mathcal{D}, \mathbf{i}_\mathcal{R}), \tag{1}$$

i.e. to compute the probabilities that mode c_i is present at an instant J_r, given the training data $\mathcal{D}$, the process knowledge $\mathbf{i}_\mathcal{R}$, and the values of the observations $\mathbf{X}^{J_1} = \mathbf{x}_{k_1}, \ldots, \mathbf{X}^{J_r} = \mathbf{x}_{k_r}$ from the process under diagnosis. Here we have used subscripts on J to denote that observation vectors from consecutive instants, and on k to enumerate the values of the observations.

4 Two Types of Knowledge

In Section 3.2 we mentioned that there may be two types of knowledge available for the fault diagnosis: training data and process specific knowledge. In this section we describe what is included in these two types of knowledge in detail.

4.1 Training Data

In many fault diagnosis problems, there is training data available. Training data consists of two ordered sets: $\mathbf{X}^{1:N} = (\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^N)$ and $\mathbf{C}^{1:N} = (C^1, C^2, \ldots, C^N)$. A realization of training data is written $\mathbf{x}^{1:N}$, $\mathbf{c}^{1:N}$, and the notation $\mathcal{D}$ is used to denote simultaneous assignments $\mathbf{C}^{1:N} = \mathbf{c}^{1:N}$ and $\mathbf{X}^{1:N} = \mathbf{x}^{1:N}$.

Training data is assumed to be collected by setting the process into different modes (or by simulation) and then recording observations. We say that training data is *experimental*, in contrast to *observational data* data which is collected by passively observing the process under operation. The fact that training data is experimental implies that training data does not have the same distribution as the data on which the fault diagnosis method will be applied to. Since the process is forced to certain modes, training data can for example not be used to learn the prior probability of the different faults, and for experimental data the following assumption holds.

Assumption 1 (Experimental Training Data)**.** *The training data does not alone provide any information about which modes that are more probable than others, i.e. it holds that*

$$p(C^j = c_i|\mathcal{D}) = p(C^j = c_i).$$

In Section 5.2 we also discuss the situation when there is also observational training data available, i.e. when Assumption 1 not holds.

4.2 Process Knowledge

Several previous algorithms for fault diagnosis rely on knowledge about that some values of the observations can impossibly occur under certain modes. We refer to this kind of information as *response information*. Response information may for example arise due to knowledge about physical properties of the process, typically expressed as qualitative or quantitative models.

Knowledge about physical properties leads to that some values of the observations can be recognized as impossible under certain modes. Consider for example the diagnosis of an electrical circuit. For the mode "open circuit" all observations of the current i_{circ} in the circuit, except the value $i_{circ} = 0$, are impossible.

Response information may also arise due to the construction of monitoring functions that are used as observations. For example, if the monitors are designed to have zero probability for false alarms, as for example in [Nyberg, 2005], then all values of the observations except the values representing "no alarm" are impossible in the fault free mode.

Many previous fault diagnosis algorithms rely on response information only, see e.g. [de Kleer and Williams, 1992, Gertler, 1998, Reiter, 1992, Nyberg, 2005].

Table 2: Example of an FSM.

	c_1	c_2	c_3
X_1^j	$\{x_{11}, x_{12}\}$	$\{x_{11}, x_{12}, x_{13}\}$	$\{x_{11}, x_{13}\}$
X_2^j	$\{x_{21}, x_{22}, x_{23}\}$	$\{x_{21}, x_{22}\}$	$\{x_{21}, x_{22}\}$

In [Gertler, 1998] response information is organized in a *residual structure*. The residual structure represents binary observations and is a matrix with one row for each observations and one column for each mode. If mode number i is known to cause observation k to alarm this is marked with a 1 in row k and column j in the residual structure. Otherwise a 0 is put in that position. In [Korbicz et al., 2004] the residual structure is called a Binary Diagnostic Matrix (BDM).

The BDM can be extended to take also multi-valued observations into account, as for example in e.g. [Daigle et al., 2006, Pulido et al., 2005, Korbicz et al., 2004]. This extension of the BDM is called a Fault Information System (FIS) or a Fault Signature Matrix (FSM). In Table 2 a part of an FSM from [Daigle et al., 2006] is given. This FSM represents two three-valued observations $X_i^j \in \{x_{i1}, x_{i2}, x_{i3}\}$, $i = 1, 2$ and three modes[4]. For each mode the possible values of the observations are listed.

In Table 3 the FSM is given for faults in the nine sensors in the engine application in Section 2.4 listed in Table 1 and for ten observations. This FSM depicts whether the observations can possible deviate from its nominal (fault free) behavior or not. That is, a 0 in the i:th column and k:th row in Table 3 means that only the nominal values of observation X_k^j are possible under mode c_i, while a "$\cdot$" means that all values of the observation X_k^j are possible. This interpretation of the FSM is similar to the one used in Structured Hypothesis Tests [Nyberg, 2005]. However, note that although there for a certain observation are 0s in two columns representing two different modes, this does not imply that that the observation has the same distribution under these two modes. It only means that no other than nominal values are possible under the corresponding modes.

5 Diagnosis Using Training Data

To do fault diagnosis, we search the probability (1) that the mode c_i is present when values $\mathbf{X}^{J_1} = \mathbf{x}_{k_1}, \ldots, \mathbf{X}^{J_d} = \mathbf{x}_{k_d}$ are observed, given training data and process knowledge. Sample numbers J_l, $l = 1, \ldots, d$ are not included in training data, i.e. $J_l \notin \{1, \ldots, N\}$.

[4]In [Daigle et al., 2006] the possible values of the observations are 0, +, and −, but here we have translated them to x_{i1}, x_{i2}, and x_{i3} respectively to fit the current notation.

Table 3: The FSM for the nine sensors and ten of the observations for the diesel engine.

	n_{eng}	n_{trb}	p_{amb}	p_{em}	p_{im}	t_{amb}	t_{im}	u_{EGR}	w_{cmp}
X_1	0	·	·	0	·	·	0	0	·
X_2	0	·	·	0	·	·	0	0	·
X_3	·	·	·	0	0	·	·	·	0
X_4	·	·	·	0	0	0	·	·	·
X_5	·	·	·	·	·	·	·	·	0
X_6	·	·	·	·	·	0	·	·	·
X_7	·	·	·	·	0	·	·	·	0
X_8	·	·	·	·	0	·	·	·	·
X_9	·	0	·	·	0	·	·	·	0
X_{10}	·	0	·	·	0	0	·	·	·

In this section we show how to compute the probabilities given training data only. In Section 5.1 we begin with the case where $d = 1$, and compute

$$p(C^{J_1} = c_i | \mathbf{X}^{J_1} = \mathbf{x}_{k_1}, \mathcal{D}, \mathbf{i}_{\mathcal{R}}). \tag{2}$$

To simplify notation, we will omit subscript 1 on J and k. In Subsection 5.3 the results are generalized to several observations. Then, in Section 6 the method is extended to also take response information into account.

5.1 One Observation

The strategy for computing the probabilities (2) when $d = 1$ and given training data only follows the same principles as in [Heckerman et al., 1995b, Pernestål and Nyberg, 2007b, Kontkanen et al., 2001]. Unlike previous works we here carefully state all assumptions, and focus on details important for the diagnosis problem. Furthermore, in contrast to [Heckerman et al., 1995b, Kontkanen et al., 2001], we consider the case of drawing conclusions about one variable (the mode variable) given the values of other variables (the observation vector), and we use experimental data. In the end of the subsection we consider the case with observational data as well.

To compute the probability (2) we begin with rewriting it by applying Bayes' theorem and using Assumption 1,

$$\begin{aligned} &p(C^J = c_i | \mathbf{X}^J = \mathbf{x}_k, \mathcal{D}) = \\ &= \frac{p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D}) p(C^J = c_i | \mathcal{D})}{\sum_{l=1}^{L} p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_l, \mathcal{D}) p(C^J = c_l | \mathcal{D})} = \\ &= \frac{p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D}) p(C^J = c_i)}{\sum_{l=1}^{L} p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_l, \mathcal{D}) p(C^J = c_l)}. \end{aligned} \tag{3}$$

The term $p(C^J = c_i)$ is the prior probability for mode c_i and is assumed to be known.

To compute the likelihood $p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D})$ in (3) we begin with partitioning training data into one part containing the training data from mode c_i and one containing the training data from all other modes. To do this, we define two index sets for each mode c_i,

$$I_{c_i} = \{l \in \{1, \ldots, N\} : C^l = c_i\},$$
$$I_{\bar{c}_i} = \{l \in \{1, \ldots, N\} : C^l \neq c_i\}.$$

Let $\mathbf{X}^{I_{c_i}}$ be a vector consisting of the elements in $\mathbf{X}^{1:N}$ with indices given by I_{c_i}, and similarly for $\mathbf{C}^{I_{c_i}}$, $\mathbf{X}^{I_{\bar{c}_i}}$, and $\mathbf{C}^{I_{\bar{c}_i}}$. We use the notation $\mathcal{D}_{c_i}$ to denote the training data from mode c_i only, i.e.

$$\mathcal{D}_{c_i} = (\mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}}, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}).$$

Example 5.1 (Notation).
To exemplify the notations, consider a training data set consisting of three samples $\mathbf{X}^{1:3} = (\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3)$, and $\mathbf{C}^{1:3} = (C^1, C^2, C^3)$ with $C^1 = c_i$, $C^2 = c_j \neq c_i$, and $C^3 = c_i$. For mode c_i we have the index sets $I_{c_i} = \{1, 3\}$ and $I_{\bar{c}_i} = \{2\}$. The training data is partitioned as $\mathbf{X}^{I_{c_i}} = (\mathbf{x}^1, \mathbf{x}^3)$, $\mathbf{C}^{I_{c_i}} = (c_i, c_i)$, and $\mathbf{X}^{I_{\bar{c}_i}} = \mathbf{x}^2$, $\mathbf{C}^{I_{\bar{c}_i}} = c_j$.

Before going into the details, we discuss two properties of the observations and the process under diagnosis.

Assumption 2 (No Memory)**.** *The probability for an observation $\mathbf{X}^j = \mathbf{x}^j$ is dependent only on the simultaneous value of the mode $C^j = c^j$, and independent of the mode at all other samples, i.e.*

$$p(\mathbf{X}^j = \mathbf{x}_k | (C^1, \ldots, C^N) = (c^1, \ldots, c^N)) = p(\mathbf{X}^j = \mathbf{x}_k | C^j = c^j).$$

In practice, Assumption 2 means that the process has no memory in the sense that an observation is independent of what might have happened in the past, and is closely related to the standard Markow assumption on dynamic processes.

Furthermore, we assume that the process under diagnosis is such that if it is known that two observations are collected from two different modes, knowing the value of one of the observations does not affect our belief in the other. When doing this assumption we rely on that when changing mode, the behavior of the system is changed.

Assumption 3 (Independent Observations)**.** *When modes are known, observations from different modes are independent*

$$p(\mathbf{X}^q = \mathbf{x}_k, \mathbf{X}^r = \mathbf{x}_l | C^q = c_i, C^r = c_j) =$$
$$= p(\mathbf{X}^q = \mathbf{x}_k | C^q = c_i, C^r = c_j) p(\mathbf{X}^r = \mathbf{x}_l | C^q = c_i, C^r = c_j),$$

where $i \neq j$.

Note that Assumption 3 does not hold for observations from the same mode. We can now state the following lemma about the likelihood in (3).

Lemma 1. *Assume that Assumptions 2 and 3 are fulfilled. Then it holds that*

$$p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D}) = p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D}_{c_i}). \tag{4}$$

Lemma 1 is proved in Appendix, and says that the probability for a certain observation, given that the mode is c_i is only dependent on the training data from that particular mode. The next step in our way to determine the likelihood is to introduce parameters $\mathbf{\Theta}_{c_i}$ with values $\theta_{c_i} = (\theta_{1c_i}, \ldots, \theta_{Kc_i})$ such that

$$p(\mathbf{X}^j = \mathbf{x}_k | C^j = c_i, \mathbf{\Theta}_{c_i} = \theta_{c_i}) = \theta_{kc_i}, \tag{5a}$$

$$\sum_{k=1}^{K} \theta_{kc_i} = 1, \quad \theta_{kc_i} > 0, \quad k = 1, \ldots, K. \tag{5b}$$

Equation (5) is a general way to parameterize a discrete distribution, since any discrete distribution can be described in this way [O'Hagan and Forster, 2004]. Furthermore, we need the following assumption on the process under diagnosis.

Assumption 4 (Independent Samples)**.** *When the parameters* $\mathbf{\Theta}_{c_i}$ *are known, observations from mode* c_i *are independent:*

$$\begin{aligned} &p(\mathbf{X}^q = \mathbf{x}_k, \mathbf{X}^r = \mathbf{x}_l | C^q = c_i, C^r = c_i, \mathbf{\Theta}_{c_i} = \theta_{c_i}) = \\ &= p(\mathbf{X}^q = \mathbf{x}_k | C^q = c_i, \mathbf{\Theta}_{c_i} = \theta_{c_i}) \times \ldots \\ &p(\mathbf{X}^r = \mathbf{x}_l | C^r = c_i, \mathbf{\Theta}_{c_i} = \theta_{c_i}). \end{aligned}$$

Assumption 4 should be interpreted as "for a given mode, the underlying process produces observation vectors that are independent". On the other hand, if the parameters are not known, two observations are in general dependent. Note that elements within an observation vector does not need to be independent even if parameters are given, but only that observation vectors from different times are independent. In our motivating example Assumption 4 is generally true, and if not true it can be obtained by using sufficiently long sampling time. We clarify the reasoning in the following toy example.

Example 5.2 (Independent trials).
Assume that there is an urn with (infinitely) many red and white balls in. Balls are drawn independently from the urn. Let R_i denote "draw number i gives a red ball". Before knowing anything about fraction of red and white balls in the urn, we assume that the probability of drawing a red ball from the urn in draw number i is $p(R_i) = 0.5$[5].

[5]This assumption is referred to as the *principle of indifference*. See e.g. [Pernestål, 2007] for a detailed discussion.

Say that the first ten trials give red balls, and denote these statements with $R_1, R_2, \ldots, R_{10}$. What is the probability that the next ball drawn is red, i.e. what is $p(R_{11}|R_1, \ldots, R_{10})$? From the first ten trials our intuition tells us that there seems to be more red balls than white balls in the urn, i.e. that $p(R_{11}|R_1, \ldots, R_{10}) > 0.5 = p(R_{11})$. The probability of trial number eleven is dependent on the result from the previous ten trials.
Now, assume that we learn that the fraction of red balls in the urn is $\theta_r = 8/10$. The knowledge of the parameter θ_r means that the probability of drawing red ball in any draw is 0.8 regardless of the balls drawn in the previous trials, $p(R_{11}|R_1, \ldots, R_{10}, \theta_r = 8/10) = p(R_{11}|\theta_r = 8/10) = 0.8$.
To summarize, the example has illustrated that before the parameters are known the trials are dependent, but when the parameters are learned the trials are independent.

In the diagnosis application, the balls in the example above represent the observations, and the distribution of balls represent the mode.

Let us now continue the computations of the likelihood. Let Ω_{c_i} be the set of all vectors θ_{c_i} that satisfies (5b). In (4), marginalize over the parameters,

$$\begin{aligned} &p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D}_{c_i}) = \\ &= \int_{\Omega_{c_i}} p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathbf{\Theta}_{c_i} = \theta_{c_i}, \mathcal{D}_{c_i}) \times \ldots \\ &f(\theta_{c_i} | C^J = c_i, \mathcal{D}_{c_i}) d\theta_{c_i}. \end{aligned} \tag{6}$$

For the first factor under the integral in (6) we can use the fact that when the parameters θ_{c_i} are known, the training data does not contribute to the belief in a certain observation. This is proved in the following lemma.

Lemma 2. *Assume that there are parameters according to* (5). *Furthermore, assume that Assumtption 4. Then it holds that*

$$\begin{aligned} &p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathbf{\Theta}_{c_i} = \theta_{c_i}, \mathcal{D}_{c_i}) = \\ &= p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathbf{\Theta}_{c_i} = \theta_{c_i}). \end{aligned}$$

The lemma is proved in Appendix. By Lemma 2 and Equation (5b) we can write the first factor under the integral in (6) as

$$\begin{aligned} &p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathbf{\Theta}_{c_i} = \theta_{c_i}, \mathcal{D}_{c_i},) = \\ &= p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathbf{\Theta}_{c_i} = \theta_{c_i}) = \theta_{k c_i}. \end{aligned} \tag{7}$$

Now, return to (6), and the second factor under the integral. By applying Bayes' theorem we can write the density as

$$f(\theta_{c_i} | C^J = c_i, \mathcal{D}_{c_i}) = \frac{p_\theta(\theta_{c_i}) f_\theta(\theta_{c_i})}{\int_{\Omega_{c_i}} p_\theta(\xi) f_\theta(\xi) d(\xi)}, \tag{8}$$

where

$$p_\theta(\xi) = p(\mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}} | \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}, \mathbf{\Theta}_{c_i} = \xi), \tag{9a}$$

$$f_\theta(\xi) = f(\xi | C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}). \tag{9b}$$

To compute the factor (9a) apply Assumption 4 and Equation (5b). This gives

$$\begin{aligned} &p(\mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}} | \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}, \mathbf{\Theta}_{c_i} = \theta_{c_i},) = \\ &= \prod_{j \in I_{c_i}} p(\mathbf{X}^j = \mathbf{x}^j | C^j = c_i, \mathbf{\Theta}_{c_i} = \theta_{c_i}) = \theta_{1c_i}^{n_{1c_i}} \dots \theta_{Kc_i}^{n_{Kc_i}}, \end{aligned} \tag{10}$$

where n_{kc_i} is the number of samples in training data from mode c_i where the observation is $\mathbf{x}_k$. From (10) we can note that the distribution of training data is directly proportional to the multinomial distribution with parameters θ_{c_i}.

The factor (9b) can be rewritten using Bayes' rule,

$$\begin{aligned} &f(\theta_{c_i} | C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}) = \\ &= \frac{p(C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}} | \mathbf{\Theta}_{c_i} = \theta_{c_i}) f(\theta_{c_i})}{p(C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}})} = \\ &= \frac{p(C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}) f(\theta_{c_i})}{p(C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}})} = f(\theta_{c_i}), \end{aligned} \tag{11}$$

where we have used that $p(C^j = c^j | \mathbf{\Theta}_{c_i} = \theta_{c_i}) = p(C^j = c^j)$ in the second equality. The result of (11) is the prior probability for the parameters.

Following [Kontkanen et al., 2001, Heckerman et al., 1995b] we assume that it is the Dirichlet distribution[6],

$$f(\theta_{c_i}) = \frac{\Gamma(\sum_{\mathbf{x}_k \in \mathbb{X}} \alpha_{kc_i})}{\prod_{\mathbf{x}_k \in \mathbb{X}} \Gamma(\alpha_{kc_i})} \prod_{\mathbf{x}_k \in \mathbb{X}} \theta_{kc_i}^{\alpha_{kc_i} - 1}, \ \alpha_{kc_i} > 0, \tag{12}$$

where $\Gamma(\cdot)$ is the gamma function, i.e. fulfills $\Gamma(n+1) = n\Gamma(n)$ and $\Gamma(1) = 1$, and the parameters $\alpha_{c_i} = (\alpha_{1c_i}, \dots, \alpha_{Kc_i})$ are given. One attractive property of Dirichlet distribution is that it is conjugate to the multinomial distribution[7] [O'Hagan and Forster, 2004], and we have noted that the distribution for the training samples is proportional to the multinomial distribution. This makes the computations particularly simple. Furthermore, the Dirichlet distribution provides the possibility of an intuitive interpretation of the parameters α_{c_i} as *hypothetical samples* in the sense that they represent samples that would have been obtained if our prior information where true. For example, if it is

[6] In fact, it can be shown that under certain, not very restrictive assumptions the Dirichlet distribution is the only possible choice for $f(\mathbf{\Theta}_{c_i} | c_i)$ [Geiger and Heckerman, 1997].

[7] The distribution $p(X = x)$ is said to be conjugate to a class of likelihood functions $p(Y = y | X = x)$ if the resulting posterior distributions $p(x|y)$ are in the same family as $p(X = x)$.

known that the value $\mathbf{x}_k$ is a-priori twice as probable as all other values this is represented by setting $\alpha_{kc_i} = 2\alpha_{jc_i}$, $j \neq k$. The amount of trust put in the prior information is regulated by choosing appropriate size of the parameters α_{c_i}; the larger value, the more confidence is put in the prior information. The interpretation of the parameters as hypothetical samples is further discussed in [Pernestål, 2007, Heckerman et al., 1995b].

The likelihood $p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D})$ can now be computed by inserting (5b), (8), (10) and (12) into (6). This gives

$$\begin{aligned} &p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D}) = \\ &= \frac{\int_{\Omega_{c_i}} \theta_{1c_i}^{n_{1c_i}+\alpha_{1c_i}-1} \dots \theta_{kc_i}^{n_{xc_i}+\alpha_{kc_i}} \dots \theta_{Kc_i}^{n_{Kc_i}+\alpha_{Kc_i}-1} d\theta_{c_i}}{\int_{\Omega_{c_i}} \theta_{1c_i}^{n_{1c_i}+\alpha_{1c_i}-1} \dots \theta_{kc_i}^{n_{xc_i}+\alpha_{kc_i}-1} \dots \theta_{Kc_i}^{n_{Kc_i}+\alpha_{Kc_i}-1} d\theta_{c_i}}, \end{aligned} \tag{13}$$

which are Dirichlet integrals of type I. We do not go into the technical details of solving the integrals, see [Pernestål and Nyberg, 2007c] for all details. We now summarize the result of the computations in this section in the following theorem.

Theorem 1. *Let $\mathbf{X}^J$ and C^J be discrete variables, and let $\{1, \dots, K\}$ be the domain of $\mathbf{X}^J$. Let $\mathcal{D}$ denote training data, and assume that Assumptions 2-4 holds. Introduce parameters $\boldsymbol{\Theta}$ according to* (5)*, and let the density $f(\boldsymbol{\Theta})$ be given by* (12)*.*

Then it holds that

$$p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D}) = \frac{n_{kc_i} + \alpha_{kc_i}}{N_{c_i} + A_{c_i}}, \tag{14}$$

where n_{kc_i} is the number of samples in training data where the observation is $\mathbf{X}^j = \mathbf{x}_k$ when $C^j = c_i$, $N_{c_i} = \sum_{k=1}^K n_{kc_i}$, and $A_{c_i} = \sum_{k=1}^K \alpha_{kc_i}$.

The posterior probability $p(C^J = c_i | \mathbf{X}^J = \mathbf{x}_k, \mathcal{D})$ can now be computed by using Theorem 1 and (3),

$$\begin{aligned} &p(C^J = c_i | \mathbf{X}^J = \mathbf{x}_k, \mathcal{D}) = \\ &= \frac{n_{kc_i} + \alpha_{kc_i}}{N_{c_i} + A_{c_i}} \frac{p(C^J = c_i)}{\sum_{j=1}^L \frac{n_{kc_j} + \alpha_{kc_j}}{N_{c_j} + A_{c_j}} p(C^J = c_j)}. \end{aligned} \tag{15}$$

Before extending the algorithm with observational data and several observations, we consider the soundness of the method presented. A diagnosis is a mode that is consistent with the observations [Hamscher et al., 1992]. In our terminology, a mode with non-zero posterior probability is a diagnosis. In the method presented above, modes can only be assigned zero posterior probability by response information. Response information includes knowledge observations that are impossible under different modes. Therefore, there is no risk for

assigning zero probability to the true underlying mode, and thus the true underlying mode will always be among those with non-zero posterior probability. This is further discussed in the comparison with related work in Section 9

5.2 Adding Observational Data

Now, let us conclude this section by instead of only experimental data consider the case where there is also observational data available, as described in Section 4.1. All the computations above in the current section holds, except the computation of $p(C^J = c_i|\mathcal{D})$ in (3). Since parts of the data is observational, the training data $\mathcal{D}$ includes information about how probable different faults are. The approach for computing the probability for the modes given the training data is similar to the one used in the previous section, and all details are given in [Pernestål and Nyberg, 2007a]. Here we are content to summarize the result:

$$p(C^J = c_i|\mathcal{D}) = \frac{N_{c_i}^{I_o} + \beta_{c_i}}{N^{I_o} + B}, \tag{16}$$

where $N_{c_i}^{I_o}$ is the number of observational training samples where the mode is c_i, N^{I_o} is the total number of observational samples, β_{c_i} are hypothetical samples describing our a priori knowledge about mode c_i, and $B = \sum_{i=1}^{L} \beta_{c_i}$.

In particular, if all training samples are experimental, (16) becomes $p(C^J = c_i|\mathcal{D}) = \beta_{c_i}/B$ which represents our a priori information about the modes formulated in terms of the parameters β_{c_i}, and is consistent with Assumption 1.

5.3 Several Observations

We will now generalize the results in the previous section, and compute the probability (1) for $d > 1$. The only difference compared to the previous section is the computation of the likelihood. To keep the notation simple we demonstrate the computations on a special case where $d = 3$. It is then straightforward to extend the computations to the case where there are more values of the observations available.

Assume that the observations $\mathbf{X}^{J_1} = \mathbf{x}_k$, $\mathbf{X}^{J_2} = \mathbf{x}_k$, and $\mathbf{X}^{J_3} = \mathbf{x}_l$ are obtained. To compute the likelihood, we use the same principle as in (6) and marginalize over the parameters Θ_{c_i},

$$\begin{aligned} &p(\mathbf{X}^{J_1} = \mathbf{x}_k, \mathbf{X}^{J_2} = \mathbf{x}_k, \mathbf{X}^{J_3} = \mathbf{x}_l|C^{J_3} = c_i, \mathcal{D}) = \\ &= \int_{\Omega_{c_i}} p_{\mathbf{X}}(\theta_{c_i}) f(\theta_{c_i}|C^{J_3} = c_i, \mathcal{D}_{c_i}) d\theta_{c_i} = \\ &= \int_{\Omega_{c_i}} \theta_{c_i k}^2 \theta_{c_i l} f(\theta_{c_i}|C^{J_3} = c_i, \mathcal{D}_{c_i}) d\theta_{c_i}, \end{aligned} \tag{17}$$

where

$$p_{\mathbf{X}}(\theta_{c_i}) = p(\mathbf{X}^{J_1} = \mathbf{x}_k, \mathbf{X}^{J_2} = \mathbf{x}_k, \mathbf{X}^{J_3} = \mathbf{x}_l | C^{J_3} = c_i, \mathbf{\Theta}_{c_i} = \theta_{c_i}, \mathcal{D}_{c_i}).$$

In (17) we have used (5) and Assumption 4 to obtain the last equality. The distribution for the parameters in (17) is computed according to Equations (8) to (12), and we obtain

$$\begin{aligned}\int_{\Omega_{c_i}} \theta_{kc_i}^2 \theta_{lc_i} f(\theta_{c_i} | C^{J_3} = c_i, \mathcal{D}_{c_i}) d\theta_{c_i} = \\ = \frac{\int_{\Omega_{c_i}} \theta_{1c_i}^{\nu_{1c_i}} \dots \theta_{kc_i}^{\nu_{kc_i}+2} \dots \theta_{lc_i}^{\nu_{lc_i}+1} \dots \theta_{\nu_{Kc_i}} d\theta_{c_i}}{\int_{\Omega_{c_i}} \theta_{1c_i}^{\nu_{1c_i}} \dots \theta_{kc_i}^{\nu_{kc_i}} \dots \theta_{lc_i}^{\nu_{lc_i}} \dots \theta_{\nu_{Kc_i}} d\theta_{c_i}},\end{aligned} \tag{18}$$

where

$$\nu_{kc_i} = n_{kc_i} + \alpha_{kc_i} - 1.$$

As in (13), the integrals in (18) are Dirichlet integrals of type 1, and can be solved analytically [Pernestål and Nyberg, 2007c]. The solution is

$$\begin{aligned}p(\mathbf{X}^{J_1} = \mathbf{x}_k, \mathbf{X}^{J_2} = \mathbf{x}_k, \mathbf{X}^{J_3} = \mathbf{x}_l | C^{J_3} = c_i, \mathcal{D}) = \\ = \frac{(n_{kc_i} + \alpha_{kc_i} + 1)(n_{kc_i} + \alpha_{kc_i})(n_{lc_i} + \alpha_{lc_i})}{(N_{c_i} + A_{c_i} + 2)(N_{c_i} + A_{c_i} + 1)(N_{c_i} + A_{c_i})}.\end{aligned}$$

With a similar derivation, the results for an arbitrary size of the set of observations can be computed. First we need some notation. Let $\mathcal{X}_{obs} \subseteq \mathbb{X}$ be the set of distinct values observed on the elements of $\mathbf{X}^{J_{1:r}} = (\mathbf{X}^{J_1}, \dots, \mathbf{X}^{J_r})$, and let D_{x_k} be the number of observations in $\mathbf{X}^{J_{1:d}}$ that take the value $\mathbf{x}_k$. In the example above we have $\mathcal{X}_{obs} = \{\mathbf{x}_k, \mathbf{x}_l\}$ and $D_{x_k} = 2$, $D_{x_l} = 1$. Using this notation, we can summarize the results in the following theorem.

Theorem 2. *Let $\mathcal{X}_{obs}$ and D_{x_k} be defined as in the previous paragraph. Let C^{J_d} and $\mathbf{X}^{J_l}$, $l = 1, \dots, d$, be discrete variables, and let $\{1, \dots, K\}$ be the domain of $\mathbf{X}^{J_l}$. Let $\mathcal{D}$ denote training data, and assume that Assumptions 1-4 holds. Introduce parameters $\mathbf{\Theta}$ according to* (5)*, and let the density $f(\mathbf{\Theta})$ be given by* (12)*. Then it holds that*

$$\begin{aligned}p(\mathbf{X}^{J_1} = \mathbf{x}_{k_1}, \dots, \mathbf{X}^{J_r} = \mathbf{x}_{k_r} | C^{J_r} = c_i, \mathcal{D}) = \\ = \frac{\prod_{\mathbf{x}_k \in \mathcal{X}_{obs}} \prod_{m=0}^{D_{\mathbf{x}_k}-1} (n_{kc_i} + \alpha_{kc_i} + m)}{\prod_{m=0}^{d-1} (N_{c_i} + A_{c_i} + m)},\end{aligned}$$

where n_{kc_i} is the number of samples in training data where the observation is $\mathbf{X}^j = \mathbf{x}_k$ when $C^j = c_i$, $N_{c_i} = \sum_{k=1}^K n_{kc_i}$, and $A_{c_i} = \sum_{k=1}^K \alpha_{kc_i}$.

6 Diagnosis Using Response Information and Data

So far we have computed the probabilities for different modes given training data only. In this section we add knowledge about the FSM to compute the probabilities. We round off with discussing the complexity of the diagnosis algorithm.

6.1 Combining Data and Response Information

Let $\mathbf{i}_{\mathcal{R}}$ denote that response information is given. Again, we first consider the case where $d = 1$. To compute the probability $p(C^J = c_i|\mathbf{X}^J = \mathbf{x}_k, \mathcal{D}, \mathbf{i}_{\mathcal{R}})$ we begin with following the steps (3) to (4) in Section 5. To determine the likelihood $p(\mathbf{X}^J = \mathbf{x}_k|C^J = c_i, \mathcal{D}_{c_i}, \mathbf{i}_{\mathcal{R}})$, we need to consider the effect of the response information more carefully. Formally, the response information means that for each mode c_i and for each observation there are (possibly empty) sets $\gamma_{jc_i} \subset \mathbb{X}_j$ of values that X_j can not take, i.e.

$$p(X_j^J = x_{jk}|C^J = c_i, \mathcal{D}, \mathbf{i}_{\mathcal{R}}) = 0, \text{ for } x_{jk} \in \gamma_{jc_i}, \tag{19}$$

We now define the set γ of impossible values of $\mathbf{X}$ as

$$\gamma_{c_i} = \{\mathbf{x}_l \in \mathbb{X} : \exists j \; x_l[j] \in \gamma_{jc_i}\},$$

i.e. if a certain value x_{jk} of observation element X_j is impossible under mode c_i, then all vectors $\mathbf{x}_l$ in which element number j takes on the value x_{jk} are also impossible.

We exemplify how the sets γ_{jc_i} and γ_{c_i} can be determined by considering the example represented by the FSM in Table 2 for the observation $\mathbf{X} = X_1$. In words, the FSM means that under the mode c_1 the observation X_1 can take on the values x_{11} and x_{12}. Under mode c_2 all values are possible, while under mode c_3 the values x_{11} and x_{13} are possible. This information gives the sets $\gamma_{1c_1} = \{3\}$, $\gamma_{1c_2} = \{\emptyset\}$, $\gamma_{1c_3} = \{3\}$. Introduce the notation $\mathbf{x}_l = x_{1l}$, $l = 1, 2, 3$. We then have $\gamma_{c_1} = \{\mathbf{x}_3\}$, $\gamma_{c_2} = \{\emptyset\}$, and $\gamma_{c_3} = \{\mathbf{x}_2\}$. The possible values of $\mathbf{X}$ under mode c_i are represented by the set $\mathbb{X}_{\mathcal{R},c_i} = \mathbb{X} \setminus \gamma_{c_i}$.

To compute the likelihood $p(\mathbf{X}^J = \mathbf{x}_k|C^J = c_i, \mathbf{\Theta}_{c_i} = \theta_{c_i}, \mathbf{i}_{\mathcal{R}})$ assume that it is parameterized by parameters θ_{c_i} as in (5b). By $\mathbf{i}_{\mathcal{R}}$ we have the following additional requirements on the parameters:

$$\theta_{kc_i} = 0, \quad \forall \mathbf{x}_k \in \gamma_{c_i}, \tag{20a}$$

$$\theta_{kc_i} > 0, \quad \forall \mathbf{x}_k \in \mathbb{X}_{\mathcal{R},c_i}. \tag{20b}$$

Each parameter θ_{kc_i} corresponds to a value in the Conditional Probability Table (CPT) describing the likelihood. This can be represented as follows.

k	$p(\mathbf{X}^j = \mathbf{x}_k \vert C^j = c_i, \theta_{c_i} = \mathbf{\Theta}_{c_i}, \mathbf{i}_\mathcal{R})$
1	θ_{1c_i}
2	θ_{2c_i}
$\vdots$	$\vdots$
K	θ_{Kc_i}

In practice the requirements (20) mean that some of the values $\mathbf{x}_k$ are impossible, i.e. that their corresponding θ_{kc_i} are identically equal to zero and their corresponding rows can be discarded from the CPT. Thus, given the knowledge $\mathbf{i}_\mathcal{R}$ the probability distribution can be described by a smaller CPT.

To compute the likelihood we follows the steps (6) to (12) in Section 5, but in (6) we integrate over the set $\Omega_{\mathcal{R}c_i}$ of parameters θ_{c_i} that fulfill (20) instead of the set Ω_{c_i}.

Consider the prior probability density function $f(\theta_{c_i}|\mathbf{i}_\mathcal{R})$. Before introducing the response information, the parameters θ_{cl_i} were assumed to be Dirichlet distributed, with density function given by (12). When adding the response information, the only knowledge that is added is that number of possible values of the observation vector is decreased. For example, this decrease in number possible values could mean that the possible values are restricted from $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ to $\{\mathbf{x}_1, \mathbf{x}_2\}$. This means that $\mathbf{i}_\mathcal{R}$ simply shrinks $\mathbb{X}$ to $\mathbb{X}_\mathcal{R}$, but changes nothing else. Let $\tilde{\theta}_{c_i}$ be the non-zero elements on of θ_{c_i} as defined in (20b). Then, the same reasoning that led to that $f(\theta_{c_i})$ in Section 5 is Dirichlet distributed, gives that $f(\tilde{\theta}_{c_i}|\mathbf{i}_\mathcal{R})$ is also be Dirichlet distributed, i.e.

$$f(\tilde{\theta}_{c_i}|\mathbf{i}_\mathcal{R}) = \frac{\Gamma(\sum_{\mathbf{x}_k \in \mathbb{X}_{\mathcal{R},c_i}} \alpha_k)}{\prod_{\mathbf{x}_k \in \mathbb{X}_{\mathcal{R},c_i}} \Gamma(\alpha_{kc_i})} \prod_{\mathbf{x}_k \in \mathbb{X}_{\mathcal{R},c_i}} \tilde{\theta}_{kc_i}^{\alpha_{kc_i}-1}, \quad \alpha_{kc_i} > 0.$$

To compute the likelihood, we note that the observation $\mathbf{x}_k \in \gamma_{c_i}$ has probability zero by (19) and the definition of γ_{c_i}. When $\mathbf{x}_k \notin \gamma_{c_i}$ we apply Theorem 1. To summarize the computations, the likelihood when response information is present is given by

$$p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D}, \mathbf{i}_\mathcal{R}) = \begin{cases} 0, & \text{if } \mathbf{x}_k \in \gamma_{c_i} \\ \frac{n_{kc_i}+\alpha_{kc_i}}{N_{c_i}+A_{c_i}} & \text{otherwise.} \end{cases} \tag{21}$$

The posterior probability for the modes given response information and training data becomes

$$p(C^J = c_i | X^J = \mathbf{x}_k, \mathcal{D}, \mathbf{i}_\mathcal{R}) = \begin{cases} 0, & \text{if } \mathbf{x}_k \in \gamma_{c_i} \\ \frac{p_i}{\sum_{j=1}^{L} p_j}, & \text{otherwise.} \end{cases} \tag{22}$$

$$\text{where } p_j = p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_j, \mathcal{D}, \mathbf{i}_\mathcal{R}) p(C^J = c_j | \mathbf{i}_\mathcal{R}). \tag{23}$$

In the case where several observations the likelihood becomes

$$p(\mathbf{X}^{J_1} = \mathbf{x}_{k_1}, \ldots, \mathbf{X}^{J_d} = \mathbf{x}_{k_d} | C^J = c_i, \mathcal{D}, \mathbf{i}_{\mathcal{R}}) = \\ = \begin{cases} 0, & \text{if } \mathcal{X}_{obs} \cap \gamma_{c_i} \neq \emptyset \\ \frac{\prod_{\mathbf{x}_k \in \mathcal{X}_{obs}} \prod_{m=0}^{D_{\mathbf{x}_k}-1} (n_{kc_i} + \alpha_{kc_i} + m)}{\prod_{m=0}^{d-1} (N_{c_i} + A_{c_i} + m)}, & \text{otherwise.} \end{cases} \tag{24}$$

6.2 Complexity of the Method

To considering the computational complexity of the method, we see from (22) that we first need to check if $\mathbf{x}_k \in \gamma_{c_j}$. This check is in worst case proportional to K, but since the FSM is typically sparse it can often be done in constant time. The next step, computing and marginalizing the likelihood is done in time proportional to L.

Considering storage of training data, the straight-forward approach to save the values n_{kc_i} for all i would require a huge but with many entries that are zero. Instead, by using the sparseness of training data the performance can be significantly decreased.

In practice, with cleverly chosen discretization of the observations, the complexity of the algorithm is linear in L. In Section 8 we will discuss some practical issues regarding discretization, observations, and modes in order to keep the complexity down.

7 Application to Diesel Engine Diagnosis

In this section we illustrate the proposed method by applying it to the diagnosis of the gas flow of a Diesel engine, using data from real driving situations.

In this section we illustrate how to apply the probabilistic method for fault diagnosis to the diesel engine described in Section 2.4 using data from real riving situations. We begin with considering diagnosis using data only, and then also add knowledge about the FSM given in Table 3. For comparison, we also perform experiments using the FSM only. To make the results easier to overview, we only present the diagnosis results from faults in the sensors

$$w_{cmp},\ u_{EGR},\ T_{im},\ \text{and}\ p_{im}. \tag{25}$$

The faults are represented by positive bias faults in representative sizes, implemented in the truck. Bias faults are chosen in the evaluation since they constitute a subset of the engine where the characteristics of the Bayesian fault diagnosis method is well illustrated, and at the same time, faults are easily implemented for experiments on this set of sensors. Although all faults considered in the present work are sensor faults, the Bayesian method applies equally well to all other kinds of faults, such as leakages, actuator faults etc. We use the

same notation to denote the sensors and the modes, i.e. fault in w_{cmp} is denoted $C^J = w_{cmp}$ etc. We also consider the fault free case, denoted NF, and the case where there is a previously unknown fault, denoted UF. The mode UF could be interpreted as "the mode is none of the other defined modes". All in all this gives us six modes.

7.1 Experimental Setup

The ten observations $\mathbf{X}_1, \ldots, \mathbf{X}_{10}$ with FSM given by Table 3 are used. For each observation we have $K_k = 3$, and the bin edges are place such that for each observation at least one of the edges is never passed in the fault free case. This selection of bin edges facilitates the use of response information. Training data used consists of 200 samples from NF, 20 samples each from w_{cmp} and p_{im}, and we use $d = 10$. The prior probabilities are set to 0.9 for the mode NF, and 0.02 for the other five modes. In order to illustrate the use of experimental data, the training data is not distributed according to the priors. All parameters representing hypothetical samples are set equal to one, $\alpha_{kc_i} = 1$.

7.2 Evaluating Diagnosis Performance

The aim of diagnosis is to provide information so that the appropriate action can be made. Therefore, the best way to evaluate a diagnosis method is to combine it with a cost function and use decision theoretical methods to compute the expected costs of different faults. Determining such a cost function is beyond the scope of the current paper. Here, the posterior probabilities for faults are the output from the method.

It is possible to define performance measures to summarize the posterior probabilities with one single figure. In [Pernestål et al., 2008] two performance measures that are relevant for diagnosis are discussed: the *logistic score* and the *percentage of correct classification*. The logistic score is commonly used in classification and statistical learning [Duda et al., 2001] and measures the ability of the method to mimic the distribution in evaluation data. However, to use it properly the evaluation data needs to be observational, which is not the case in the current application. The percentage of correct classification measures the probability of doing a correct choice if the mode with largest probability is chosen as the true mode. It reflects the performance of the fault diagnosis method together with the naive troubleshooting strategy "check most probable fault first".

In the current paper we have chosen to present the average posterior probabilities since they provide the more details about the behavior of the algorithm. To evaluate the diagnosis performance, $N_{eval} = 100$ evaluation samples from each of the six modes are considered. The mode UF is represented by a previ-

ously unknown fault, from which neither training data nor response information exist.

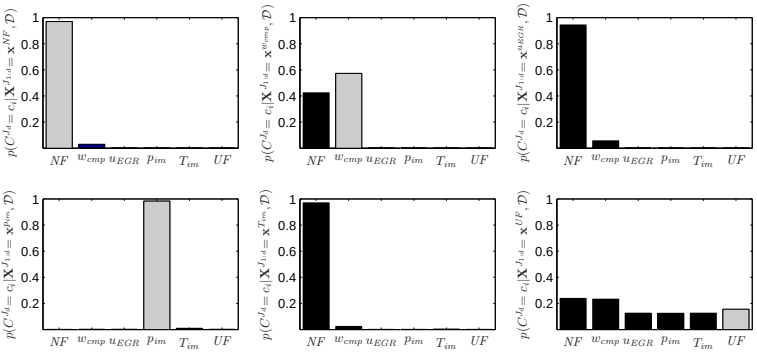
7.3 Fault Diagnosis Using Training Data Only and Response Information Only

We compute the probabilities for faults using data only by applying Theorem 2 and the prior probabilities given above. The results are shown in Figure 4a. In each subfigure, evaluation data is collected from different modes. The true underlying mode is depicted in gray. Remember that the diagnosis is based on training samples from the modes NF, w_{cmp}, and p_{em} only. These three modes are isolated with high precision, while the diagnosis of the other modes is worse. From the modes u_{EGR}, T_{im}, and UF on the other hand, no training data exists. The reason for the dominance of the mode NF when evaluation data comes from u_{EGR} and T_{im} is the high prior probability of the mode NF in combination with that for several observations the behavior these two modes gives similar values as the mode NF. The values of the observations from mode UF differs significantly from observations from the other modes. Therefore, the probability mass is almost equally distributed when evaluation data is from this mode.

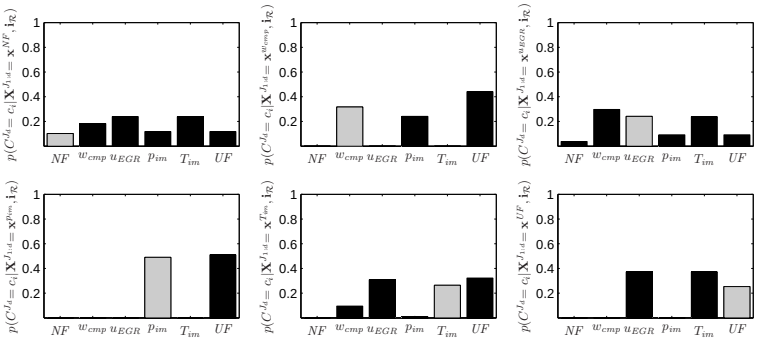
For comparison, the diagnosis result using response information only is shown in Figure 4b. For this case, the probability mass is more equally distributed over several modes.

7.4 Fault Diagnosis Using Response Information and Training Data

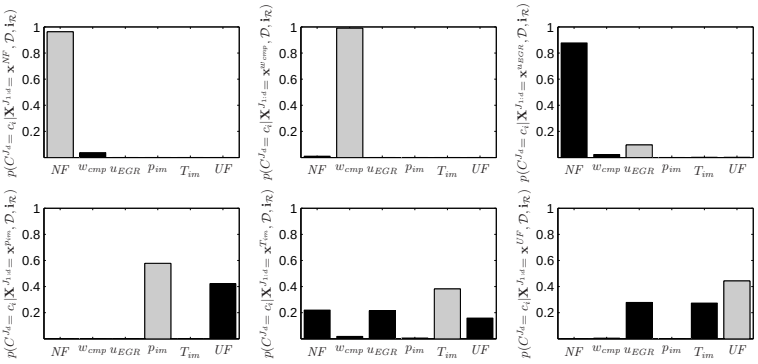
We now add response information to the fault diagnosis, and compute the posterior probabilities by using (24) and the priors given above. By using the FSM in Table 3 we can form the sets $\mathbb{X}_{\mathcal{R},c_i}$ for each mode. For the mode UF no response information exists, and all values are possible. This gives $\mathbb{X}_{\mathcal{R},UF} = \emptyset$. In Figure 4c are the results from the diagnosis plotted. The true underlying mode is marked with a gray bar. Comparing with the diagnosis result using training data only, we see that the diagnosis performance has improved significantly for the modes UF and T_{im}. Also, for the mode u_{EGR}, the diagnosis performance is slightly improved. However, the mode u_{EGR} is difficult to isolate since its influence on the observations is very small, see [Molin and Hansen, 2006]. To summarize the application results, we can see from Figure 4 that the traditional methods using either response information only or training data only do not perform well on the current application. However, by combining both training data and response information the diagnosis performance is significantly improved.



(a) Training data



(b) Response information



(c) Response information and training data

Figure 4: The average probability assigned to each mode when evaluation data comes from different modes, using response information and/or training data. The true underlying mode is marked with gray.

8 Discussion About Practical Issues

In the Bayesian method for fault diagnosis introduced in Sections 5 and 6, there are several design choices to be made. In this section we discuss four of them: the selection of modes, the selection of observations, the discretization of the observations, and the collection of training data.

8.1 Choice of Modes

To use the proposed diagnosis method, the modes to diagnose must be specified. A simple, but naive specification of modes is to use one mode for each possible fault or combination of faults. However, such a specification would potentially lead to unnecessary detailed diagnosis and, if considering many multiple faults, an explosion in the number of modes considered. In particular, from (3) it can be seen that the likelihood must be computed for each mode. Therefore, besides the practical issue of specifying modes, it is also crucial to keep the number of modes as small a possible. In this section we go through three ways of lumping modes together to decrease the computational and storage efforts needed.

First, as suggested for example in [Cordier et al., 2007, Pernestål, 2007], it might not be necessary to distinguish between some modes. We assume that the output from the diagnosis is to be used to decide the best action to perform. If two modes (or faults) always lead to the same action they can be grouped into one lumped mode. In [Pernestål, 2007] such a lumped mode a is referred to as a *Diagnosed Mode*, while in [Cordier et al., 2007] the are called *Macrofaults*. Which modes to lump can for example be found by using FMEA/FMECA analysis [Stamatis, 1995]. The FSM column for the lumped mode is obtained by, for each observation, taking the intersection of impossible values. How to define which modes that should be contained in each lumped mode is beyond the scoop of the present paper. Here we are confident to note that the probabilistic framework presented in Sections 5 and 6, and all computations are the same with the modes exchanged to lumped modes and we will simply use the term "mode" also for lumped modes.

Second, we note that for all modes c_i and c_j for which

$$n_{kc_i} + \alpha_{kc_i} = n_{kc_j} + \alpha_{kc_j} \tag{26}$$

the likelihood is equal. Thus, for all modes that satisfy (26), the computations need only to be performed once and the result can be reused.

Third, taking this reasoning one step further, we note that modes with equal FSM columns and equal distribution of training data (e.g. no training data), can only be distinguished by their prior probabilities. Therefore, such modes can be grouped into one mode during the computations. When combining more and more faults in each mode, the columns in the FSM will have more and more '·'-entries and thus become more and more equal. Furthermore, there will in

general be no training data from modes with several multiple faults. Thus, these modes can only be distinguished by their prior probability. Considering multiple fault modes, it is likely that several such modes have the same counter action, and thus they can be lumped into one mode. Thus, the seemingly large restriction that only predetermined modes can be diagnosed is not really a restriction in practice.

The number of modes can be also be reduced by considering the physical or logical structure of the process, and, if possible, divide it into independent subprocess. Then, one mode variable is used for each subsystem. For example, in the automotive engine the gas flow subprocess can be considered as independent from the fuel injection subprocess. A further approach to keep number of modes small is to use a hierarchical approach, and first compute the probability of groups of modes, for example corresponding to sub-processes of the process under diagnosis. When the probability for one group of modes is sufficiently large, the probability of the modes in this group can be computed.

8.2 Discretization

Another important design choice that affect the result of the diagnosis method is the discretization of the observations into a number of bins: how many bins to use, and where to put the bin edges. The task of discretization is a research area on its own, and choosing the optimal discretization is beyond the scope of the current paper. Instead, we give some practical advises concerning discretization.

The question of the number of bins is related to the problem of choosing a set of residuals to use discussed in Section 8.4 in the sense that a larger number of bins gives a higher resolution but requires more training data [MacKay, 2003]. Often, the number of bins can be relatively small, since in many situations it is enough to know the direction of deviation ("positive" or "negative") or the magnitude ("small" or "large") from the nominal value of the observation. Using the sensor readings instead of monitoring functions would typically require finer discretization. Where to put the bin edges depends, of course, on how the data is distributed. One strategy that is used for example in [Nyberg, 2005] is to place the bin edges such that the probability of false alarm is approximately zero. Other strategies that might be useful when using histograms to represent probability distributions as in the current paper is for example Minimum Description Length (MDL) histograms [Kontkanen and Myllymäki, 2007] or methods based on Maximum Entropy methods [Johansson, 2005].

8.3 Selection of Training Data

The performance of the diagnosis method is dependent on the training data used to learn its parameters. When learning the parameters in the diagnosis method, all data available should be used. In the current paper we consider

both experimental and observational data. While observational data is simply collected by observing the system, experimental data is obtained by actively implementing faults before collection of data. Here we have a design choice: which faults to implement and collect data from?

Ignoring the fact that some faults are not possible to implement, we give two guidelines. First, data should be collected from modes that are important to diagnose with high precision. Collecting data from these modes improve the diagnosis of them. Second, data should be collected from modes that have similar columns in the FSM. Modes with similar (equal) columns are difficult (impossible) to distinguish by using the FSM only. In these cases data will help distinguishing between the faults.

8.4 Selection of Observations

In real applications, at least to our experience, there will sometimes be more monitors available than can be run. For example, in the diesel engine studied in the current paper it is possible to automatically generate more than 60 possible monitoring functions only for one specific subprocess. At least in a structural sense, these monitors are redundant. The limited computational capacity restricts the number of monitors that can possibly be executed. Furthermore, since the amount of training data is finite, and in fact often very limited from fault modes, the diagnosis result may even be better if only a few of the observations are selected, see e.g. [MacKay, 2003]. Therefore, it is indeed an interesting question how to choose the best subset of monitors (observations) to use for diagnosis.

Observation selection (often called feature selection) is a research area on its own, and finding an set of monitors that guarantees the best possible diagnosis performance is beyond the scope of the current paper. However, a general recommendations is to consider structural diagnosis information about which observations that possibly can detect certain faults (e.g. FSM knowledge) together with evaluation on data (for example by cross validation and the performance measures suggested in Section 7.2). A set of observations that provides full isolability in a structural sense is not necessarily the best when data is added. Methods for observation selection is further discussed in [Pernestål, 2007, Pernestål et al., 2008].

9 Relation to Previous Works

To improve understanding of the diagnosis method presented here we now discuss the relations between the current method and previous model-based methods for fault diagnosis. In the comparison we have chosen the Sherlock method [de Kleer and Williams, 1992] from the AI community, the method

based on structured residuals [Gertler, 1998] from the automatic control (FDI) community, and a general model-based probabilistic method. These methods are chosen since they are state of the art, and many other fault diagnosis methods rely on one or several of these three approaches.

9.1 Relation to Sherlock

In the AI field many fault diagnosis methods rely on the computation of sets of modes that are consistent with the current observations [de Kleer and Williams, 1992, Reiter, 1992]. One of these methods is the Sherlock algorithm presented in [de Kleer and Williams, 1992]. The Sherlock algorithm also includes parts for determining the next best observation to perform, but here we consider only the fault localization part of Sherlock. Sherlock performs probabilistic computations based on response information only. It is assumed that all observations are independent, i.e. that $p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathbf{i}_\mathcal{R}) = \prod_{l=1}^{R} p(X_l^J = x_k[l] | C^J = c_i, \mathbf{i}_\mathcal{R})$, and the following probabilities are used:

$$p(X_l^J = x_k[l] | C^J = c_i, \mathbf{i}_\mathcal{R}) = \begin{cases} 0, & \text{if } x_k[l] \text{ is surely} \\ & \textit{inconsistent} \text{ with } c_i \\ 1, & \text{if } x_k[l] \text{ is surely} \\ & \textit{consistent} \text{ with } c_i \\ \frac{1}{K_l}, & \text{otherwise.} \end{cases} \tag{27}$$

Recall from Section 3.1 that K_l is the number of possible values of X_l. With "surely inconsistent" we mean that the observation is impossible, i.e. that $x_k[l] \in \gamma_{lc_i}$. With "surely consistent" we mean that $\gamma_{lc_i} = \mathbb{X} \setminus \{x_k[l]\}$, i.e. that $x_k[l]$ is the only possible value of X_l^J when the mode is $C^J = c_i$. For the complete observation vector, (27) gives

$$p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathbf{i}_\mathcal{R}) = \begin{cases} 0, & \text{if } \mathbf{x}_k \in \gamma_{c_i} \\ \prod_{l \in I_{c_i}^{Cons}} \frac{1}{K_l}, & \text{otherwise,} \end{cases} \tag{28}$$

where $I_{c_i}^{Cons}$ is the set of indices of the observations which are neither surely consistent nor surely inconsistent with the mode c_i.

When there is no training data in the method developed in Section 6, we obtain the likelihood

$$p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathbf{i}_\mathcal{R}) = \begin{cases} 0, & \text{if } \mathbf{x}_k \in \gamma_{c_i} \\ \frac{\alpha_{kc_i}}{A_{c_i}}, & \text{otherwise.} \end{cases} \tag{29}$$

With $\alpha_{kc_i} = 1$ for $\mathbf{x}_k \notin \gamma_{c_i}$, $\alpha_{kc_i} = 0$ for $\mathbf{x}_k \in \gamma_{c_i}$, and when there is no training data available we have $A_{c_i} = \prod_{l \in I_{c_i}^{Cons}} \frac{1}{K_l}$, and the current method becomes the

same as Sherlock. However, by using our method it is possible to improve the diagnosis by adding training data.

Another interesting feature with the current method is that it facilitates the use of knowledge about that some values of the observations are a priori more probable than other methods. Consider for example a system consisting of electrical circuits. If the system consists mainly of "or"-gates, in the fault free case an observation vector consisting mostly of 1:s is a priori more probable than an observation vector consisting mostly of 0:s. The reason is that out of the four possible inputs to an "or"-gate, (1,1), (1,0), (0,1), (0,0), only the last one gives a 0 as output. In this case, Sherlock would overestimate the likelihood for observation vectors with a lot of 0:s and underestimate the likelihood for observation vectors with lot of 1:s in the fault free case, and produce erroneous results. This effect is noticed [de Kleer, 2006], but no solution to the problem is given. In our current method it is easily implemented by adjusting the hypothetical samples.

9.2 Relation to Structured Residuals

Structured Residuals is one approach to fault diagnosis that is the basis in many algorithms in the automatic control community. In structured residuals, the fault diagnosis rely on an FSM, where each row represents a residual, and each column represents a fault, [Gertler, 1998, Patton et al., 2000]. In the FSM a 0 in the jth row and ith column marks that observation j is not sensitive to mode c_i, and a 1 means that it is sensitive [Patton et al., 2000]. Fault diagnosis is performed by matching the current values of the residuals, i.e. the current observation vector, with the columns in the FSM.

One problem with fault diagnosis as defined above is that it requires 100% response of the observations and 0% false alarm, otherwise the diagnosis will be erroneous. To solve this, some solutions are suggested, for example by computing the (Hamming) distance between the current observation vector and the columns in the FSM, or by using fuzzy logic [Patton et al., 2000, Korbicz et al., 2004]. Another approach is to relax the ones in the FSM to mean that the observation *may* be respond to that fault as in Structured Hypothesis Testing (SHT) [Nyberg, 2005]. In the following small example we compare the current probability based method with the SHT method.

Example 9.3 (Comparison with SHT).
Consider the case with two binary observations, $X_j \in \{0,1\}$, $j = 1,2$, three possible faults, and with FSM represented by

	c_1	c_2	c_3
X_1	X	0	0
X_2	X	X	X

(30)

where an X in the jth row and the ith column means that observation X_j may respond in mode c_i. Let $X_j = 1$ mean that observation j has responded, and that the observation vector $(x_1, x_2) = (0, 1)$ is obtained. Using the SHT method, all three faults are presented as possible to be present.
In the current method, the probabilities are computed by using (22). Assuming that all faults are a priori equally probable we obtain

$$\begin{aligned} &p(C^J = c_i|(X_1, X_2)^J = (0, 1), \mathcal{D}, \mathbf{i}_\mathcal{R}) = \\ &= \frac{1}{\rho}\frac{n_{(01)c_i} + \alpha_{(01)c_i}}{N_{c_i} + A_{c_i}} p(C^J = c_i|\mathbf{i}_\mathcal{R}), \\ &\text{where} \quad \rho = \sum_{i=1}^{3} \frac{n_{(01)c_i} + \alpha_{(01)c_i}}{N_{c_i} + A_{c_i}} p(C^J = c_i|\mathbf{i}_\mathcal{R}), \end{aligned}$$

where $n_{(01)c_i}$ is the number of training samples from mode c_i where the observation vector is $\mathbf{X}^j = (0, 1)$ and similar for the hypothetical samples $\alpha_{(01)c_i}$.
If there is no training data available, and all parameters α_i are set to be equal, all three faults are assigned the equal probability and the result from the current method is the same as from SHT (if SHT is interpreted in probability terms). On the other hand, if there is training data available, it can be used to improve the diagnosis in the current method. In particular, note that training data may make it possible to distinguish between the modes c_2 and c_3, while these two fault can never be distinguished by using SHT only.

9.3 Relation to Model-Based Probabilistic Methods

Probabilistic reasoning for fault diagnosis is successfully used in several previous works. Common in these references are that they aim at computing probabilities for faults by using some kind of probabilistic model, S, that describes the probabilistic relations between observations and faults. The probabilistic model is often a Bayesian Network. This probabilistic model may be estimated from data, as e.g. in [Verron et al., 2007, Pernestål et al., 2006, Pernestål et al., 2008], or set up using expert knowledge as e.g. in [Schwall and Gerdes, 2002, Lerner et al., 2000, Narasimhan and Biswas, 2007].

In the current work, no explicit probabilistic model is used. Instead the probability computations are performed using training data and possibly the response information, and we compute the probability $p(C^J = c_i|\mathbf{X}^J = \mathbf{x}_k, \mathcal{D}, \mathbf{i}_\mathcal{R})$ directly. To study the difference between the previous and the present method, marginalize over all possible probabilistic models,

$$p(C^J = c_i|\mathbf{X}^J = \mathbf{x}_k, \mathcal{D}, \mathbf{i}_\mathcal{R}) = \int p(C^J = c_i|\mathbf{X}^J = \mathbf{x}_k, S = s, \mathbf{i}_\mathcal{R}) f(s|\mathcal{D}, \mathbf{i}_\mathcal{R}) ds.$$

Here, we have used two assumptions: (1) that when the model is known, $\mathcal{D}$ does not provide any further information about C^J; and (2) that $\mathbf{X}^J$ alone (without the corresponding C^J) does not provide any information about S.

With the approximation

$$f(s|\mathcal{D}, \mathbf{i}_\mathcal{R}) = \delta(s - s_0), \tag{31}$$

where $\delta(\cdot)$ is the probability distribution with all probability mass centered in the point s_0, we have that

$$p(C^J = c_i|\mathbf{X}^J = \mathbf{x}_k, \mathcal{D}, \mathbf{i}_\mathcal{R}) = p(C^J = c_i|\mathbf{X}^J = \mathbf{x}_k, S = s_0, \mathbf{i}_\mathcal{R}).$$

The approximation (31) may be good when there is one model that is far more probable than the others. However, this is often not the case in fault diagnosis due to the lack of training data from some modes.

9.4 Relation to Bayesian Networks

A Bayesian network (BN) is a directed acyclic graph representing the joint probability distribution over a set of variables, see e.g. [Jensen and Nielsen, 2007] for an extensive description of BNs. In the BN, each node represents a variable, arcs between nodes represent probabilistic dependencies between nodes (variables), and each node is equipped with the conditional probability table (CPT)[8] for its variable, given its parents. The diagnosis method presented here could be represented by a BN with one node for the mode and one (multidimensional) node for the observations $\mathbf{X}$, see Figure 5 (a). The CPT for $\mathbf{X}$ is given by (21), and the probability for the mode is given by (16).

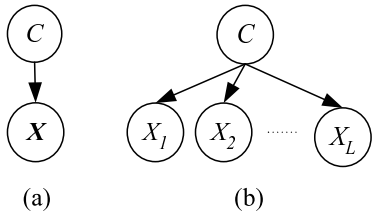


Figure 5: Two BNs that could be used for diagnosis.

It is tempting to let each observation be represented by a single node and to assume that the nodes are independent, see Figure 5 (b). This structure of BN is often referred to as a naive Bayes net (NB). In the NB, the CPTs are significantly smaller than the full CPT used in Figure 5 (a). The CPTs in the NB are assigned using (14) with a one-dimensional observation $\mathbf{X}^J =$

[8] For continuous variables the conditional probability density is used.

X_l^J. One important difference between NB and the current method is that in the NB it is assumed that observations are independent. This is not true in the diesel engine application, since there are several unmodeled effects that affect the observation. The erroneous assumption of independence affects the diagnosis performance, see e.g. [Pernestål et al., 2006]. There are other possible structures, such as the Tree Augmented Naive Bayesian network (TAN) which allows slightly more complicated dependencies than NB, but still may suffer from to strong independence assumptions.

In [Pernestål et al., 2008] different state-of-the-art methods are applied to learning BNs for diagnosis, and three main conclusions are presented. First, most learning algorithms for BNs assumes observational data. In our application, data is typically experimental, and this may lead to erroneous BNs. Second, the lack of data from some (actually most) of the modes is not handled in the state-of-the-art learning algorithms. Third, these state-of-the-art methods does not take response information into account.

10 Conclusion

A new Bayesian method for fault diagnosis has been proposed. The aim of the work has been to design a generic fault diagnosis method, applicable to real world automotive systems. The work has been motivated by the diagnosis of an automotive diesel engine, and the characteristics of the application has been carefully studied.

The new method combines training data and process knowledge in terms of an FSM, and computes the probabilities for faults given all available information. The probabilities can then be combined with cost functions in the decision theoretic framework, to determine the best action to perform to take the process back to a safe and efficient operating mode.

With carefully chosen design parameters, for example by utilizing lumping of modes, the proposed method has low complexity.

Appendix

Proof of Lemma 1. Begin with applying the product rule of probabilities,

$$\begin{aligned} &p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D}) = \\ &= p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathbf{X}^{1:N} = \mathbf{x}^{1:N}, \mathbf{C}^{1:N} = \mathbf{c}^{I_{c_i}}) = \\ &= \frac{p(\mathbf{X}^J = \mathbf{x}_k, \mathbf{X}^{1:N} = \mathbf{x}^{1:N} | C^J = c_i, \mathbf{C}^{1:N} = \mathbf{c}^{1:N})}{p(\mathbf{X}^{1:N} = \mathbf{x}^{1:N} | C^J = c_i, \mathbf{C}^{1:N} = \mathbf{c}^{1:N})}. \end{aligned} \tag{32}$$

By using Assumptions 2 and 3 we have that

$$\begin{aligned} &p(\mathbf{X}^{1:N} = \mathbf{x}^{1:N} | C^J = c_i, \mathbf{C}^{1:N} = \mathbf{c}^{1:N}) = \\ &= p(\mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}} | C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}) \times \dots \\ &p(\mathbf{X}^{I_{\bar{c}_i}} = \mathbf{x}^{I_{\bar{c}_i}} | C^J = c_i, \mathbf{C}^{I_{\bar{c}_i}} = \mathbf{c}^{I_{\bar{c}_i}}), \end{aligned} \tag{33}$$

and

$$\begin{aligned} &p(\mathbf{X}^J = \mathbf{x}_k, \mathbf{X}^{1:N} = \mathbf{x}^{1:N} | C^J = c_i, \mathbf{C}^{1:N} = \mathbf{c}^{1:N}) = \\ &= p(\mathbf{X}^J = \mathbf{x}^J, \mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}} | C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}) \times \dots \\ &p(\mathbf{X}^{I_{\bar{c}_i}} = \mathbf{x}^{I_{\bar{c}_i}} | C^J = c_i, \mathbf{C}^{I_{\bar{c}_i}} = \mathbf{c}^{I_{\bar{c}_i}}). \end{aligned} \tag{34}$$

By inserting (33) and (34) in (32), and then applying the product rule of probabilities we have

$$\begin{aligned} &\frac{p(\mathbf{X}^J = \mathbf{x}_k, \mathbf{X}^{1:N} = \mathbf{x}^{1:N} | C^J = c_i, \mathbf{C}^{1:N} = \mathbf{c}^{1:N})}{p(\mathbf{X}^{1:N} = \mathbf{x}^{1:N} | C^J = c_i, \mathbf{C}^{1:N} = \mathbf{c}^{1:N})} = \\ &= \frac{p(\mathbf{X}^J = \mathbf{x}_k, \mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}} | C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}})}{p(\mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}} | C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}})} \times \dots \\ &\frac{p(\mathbf{X}^{I_{\bar{c}_i}} = \mathbf{x}^{I_{\bar{c}_i}} | C^J = c_i, \mathbf{C}^{I_{\bar{c}_i}} = \mathbf{c}^{I_{c_i}})}{p(\mathbf{X}^{I_{\bar{c}_i}} = \mathbf{x}^{I_{\bar{c}_i}} | C^J = c_i, \mathbf{C}^{I_{\bar{c}_i}} = \mathbf{c}^{I_{c_i}})} = \\ &= \frac{p(\mathbf{X}^J = \mathbf{x}_k, \mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}} | C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}})}{p(\mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}} | C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}})} = \\ &= p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}}, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}). \end{aligned}$$

With the notation $\mathcal{D}_{c_i} = (\mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}}, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}})$ for the training data from mode c_i. Then we can write

$$p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}}, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}) = p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D}_{c_i})$$

and the Lemma is proved. □

Proof of Lemma 2. Apply the product rule of probabilities

$$\begin{aligned} &p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D}_{c_i}, \mathbf{\Theta}_{c_i} = \theta_{c_i}) = \\ &= \frac{p(\mathbf{X}^J = \mathbf{x}_k, \mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}} | C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}, \mathbf{\Theta}_{c_i} = \theta_{c_i})}{p(\mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}} | C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}, \mathbf{\Theta}_{c_i} = \theta_{c_i})}. \end{aligned} \tag{35}$$

By using Assumtpions 2 and 4 we have that

$$\begin{aligned} &p(\mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}} | C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}, \mathbf{\Theta}_{c_i} = \theta_{c_i}) = \\ &= \prod_{j \in I_{c_i}} p(\mathbf{X}^j = \mathbf{x}^j | \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}, \mathbf{\Theta}_{c_i} = \theta_{c_i}) \end{aligned} \tag{36}$$

and

$$p(\mathbf{X}^J = \mathbf{x}_k, \mathbf{X}^{I_{c_i}} = \mathbf{x}^{I_{c_i}} | C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}, \mathbf{\Theta}_{c_i} = \theta_{c_i}) =$$
$$= p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}, \mathbf{\Theta}_{c_i} = \theta_{c_i}) \times \ldots$$
$$\prod_{j \in I_{c_i}} p(\mathbf{X}^j = \mathbf{x}^j | \mathbf{C}^{I_{c_i}} = \mathbf{c}^{I_{c_i}}, \mathbf{\Theta}_{c_i} = \theta_{c_i}) \quad (37)$$

By inserting (36) and (37) into (35) we obtain

$$p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathcal{D}_{c_i}, \mathbf{\Theta}_{c_i} = \theta_{c_i}) = p(\mathbf{X}^J = \mathbf{x}_k | C^J = c_i, \mathbf{\Theta}_{c_i} = \theta_{c_i})$$

and the lemma is proved. □

References

[Alonso-González et al., 2008] Alonso-González, C. J., Rodríguez, J. J., Prieto, O. J., and Pulido, B. (2008). Machine learning and model based diagnosis using possible conflicts and system decomposition. In *Proceedings The 19th International Workshop on Principles of Diagnosis*.

[Becraft et al., 1991] Becraft, W. R., Lee, P. L., and Newell, R. B. (1991). Integration of neural networks and expert systems for process fault diagnosis. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 832–837.

[Bregon et al., 2007] Bregon, A., Pulido, B., Simon, A., Moro, Q. I., Prieto, O.-J., Rodriguez, J. J., and Alonso, C. (2007). Focusing Fault Localization in Model-based Diagnosis with Case-based Reasoning. In *Proceedings of the European Control Cinference*.

[Cascio et al., 1999] Cascio, F., Console, L., Guagliumi, M., Osella, M., Panati, A., Sottano, S., and Dupre, D. T. (1999). Generating on-borad diagnostics of dynamic automotive systems based on qualitative models. *AI Communications*, 12:33–43.

[Cordier et al., 2007] Cordier, M.-O., Pencole, Y., Trave-Massuyes, L., and Vidal, T. (2007). Self-healability = diagnosability + repairability. In *Proceedings of 18th International Workshop on Principles of Diagnosis (DX 07)*, pages 251–258.

[Daigle et al., 2006] Daigle, M., Koutsoukos, X., and Biswas, G. (2006). Multiple Fault Diagnosis in Complex Physical Systems. In *Proceedings of 17th International Workshop on Principles of Diagnosis (DX 06)*.

[de Kleer, 2006] de Kleer, J. (2006). Getting the Probabilities Right for Measurement Selection. In *Proceedings of 17th International Workshop on Principles of Diagnosis (DX 06)*, pages 141– 146.

[de Kleer and Williams, 1992] de Kleer, J. and Williams, B. C. (1992). Diagnosis with Behavioral Modes. In *Readings in Model-based Diagnosis*, pages 124–130, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Duda et al., 2001] Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley, New York.

[Fouladirad and Nikiforov, 2005] Fouladirad, M. and Nikiforov, I. (2005). Optimal Statistical Fault Detection with Nuisance Parameters. *Automatica*, 41:1157 – 1171.

[Geiger and Heckerman, 1997] Geiger, D. and Heckerman, D. (1997). A Characterization of the Dirichlet Distribution Through Global and Local Independence. *The Annals of Statistics*, 25(3):1344–1360.

[Gertler et al., 1995] Gertler, J., Costin, M., Fang, X., Kowalczuk, Z., Kunwer, M., and Monajemy, R. (1995). Model based diagnosis for automotive engines - algorithm development and testing on a production vehicle. *IEEE Transactions on Control Systems Technology*, 3(1):61–69.

[Gertler, 1998] Gertler, J. J. (1998). *Fault Detection and Diagnosis in Engineering Systems*. Marcel Decker, New York.

[Hamscher et al., 1992] Hamscher, W., Console, L., and deKleer, J. (1992). *Readings in Model-based Diagnosis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Heckerman et al., 1995a] Heckerman, D., Breese, J. S., and Rommelse, K. (1995a). Decision-theoretic troubleshooting. *Communications of the ACM*, 38(3):49–57.

[Heckerman et al., 1995b] Heckerman, D., Geiger, D., and Chickering, D. M. (1995b). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20(3):197–243.

[Jaynes, 2001] Jaynes, E. T. (2001). *Probability Theory - the Logic of Science*. Camebridge University Press, Cambridge.

[Jensen and Nielsen, 2007] Jensen, F. V. and Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs*. Springer.

[Johansson, 2005] Johansson, M. (2005). Approximate bayesian inference by adaptive quantization of the hypothesis space. In *Proceedings of 25th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (MaxEnt 2005)*.

[Kontkanen and Myllymäki, 2007] Kontkanen, P. and Myllymäki, P. (2007). MDL histogram density estimation. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 2007)*.

[Kontkanen et al., 2001] Kontkanen, P., Myllymäki, P., Silander, T., Tirri, H., and Grünwald, P. (2001). Comparing predictive inference methods for discrete domains. In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, pages 233–238.

[Korbicz et al., 2004] Korbicz, J., Koscielny, J. M., Kowalczuk, Z., and Cholewa, W. (2004). *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Springer, Berlin, Germany.

[Langseth and Jensen, 2002] Langseth, H. and Jensen, F. V. (2002). Decision theoretic troubleshooting of coherent systems. *Reliability Engineering & System Safety*, 80(1):49–62.

[Lee et al., 2005] Lee, B., Guezennec, Y., and Rizzoni, G. (2005). Model-based fault diagnosis of spark-ignition direct-injection engine using nonlinear estimations. *SAE Transactions*, 114(1):190–200.

[Lee et al., 2007] Lee, G., Bahri, P., Shastri, S., and Zaknich, A. (2007). A Multi-Category Decision Support System Framework for the Tennessee Eastman Problem. In *Proceedings of the European Control Conference (ECC 07)*.

[Lerner et al., 2000] Lerner, U., Parr, R., Koller, D., and Biswas, G. (2000). Bayesian Fault Detection and Diagnosis in Dynamic Systems. In *AAAI/IAAI*, pages 531–537.

[MacKay, 2003] MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

[Molin and Hansen, 2006] Molin, J. and Hansen, J. (2006). Design and Evaluation of an Automatically Designed Diagnosis System. Master's thesis, Linköping University.

[Narasimhan and Biswas, 2007] Narasimhan, S. and Biswas, G. (2007). Model-Based Diagnosis of Hybrid Systems. *IEEE Transactions on Man, Systems and Cybernetics – part A*, 37(3):348–361.

[Nyberg, 2005] Nyberg, M. (2005). Model-Based Diagnosis of an Automotive Engine Using Several Types of Fault Models. *IEEE Transactions on Control Systems Technology*, 10(5):679–689.

[O'Hagan and Forster, 2004] O'Hagan, A. and Forster, J. (2004). *Kendall's Advanced Theory of Statistics*. Arnold, London.

[Patton et al., 2000] Patton, R. J., Frank, P. M., and Clark, R. N. (2000). *Issues of Fault Diagnosis for Dynamic Systems*. Springer, New York.

[Pernestål, 2007] Pernestål, A. (2007). *A Bayesian Approach to Fault Isolation with Application To Diesel Engine Diagnosis*. Lic. Thesis, Royal Institute of Technology, Stockholm, Sweden.

[Pernestål and Nyberg, 2007a] Pernestål, A. and Nyberg, M. (2007a). Experimental and Observational Data in Learning for Bayesian Inference. Technical Report LiTH-ISY-R-2834, ISY, Linköping University.

[Pernestål and Nyberg, 2007b] Pernestål, A. and Nyberg, M. (2007b). Probabilistic Fault Isolation Based on Incomplete Training Data with Application to an Automotive Engine. In *Proceedings of the European Control Conference (ECC 07)*.

[Pernestål and Nyberg, 2007c] Pernestål, A. and Nyberg, M. (2007c). Using Data and Prior Information in Bayesian Classification. Technical Report LiTH-ISY-R-2811, ISY, Linköping University.

[Pernestål et al., 2006] Pernestål, A., Nyberg, M., and Wahlberg, B. (2006). A Bayesian Approach to Fault Isolation with Application to Diesel Engine Diagnosis. In *Proceedings of 17th International Workshop on Principles of Diagnosis (DX 06)*, pages 211–218.

[Pernestål et al., 2008] Pernestål, A., Wettig, H., Silander, T., Nyberg, M., and Myllymäki, P. (2008). A bayesian approach to learning in fault isolation. In *Proceedings of the 19th International Workshop on Principles of Diagnosis*.

[Pulido et al., 2005] Pulido, B., Puig, V., Escobet, T., and Quevedo, J. (2005). A New Fault Localization Algorithm that Improves the Integration Between Fault Detection and Localization in Dynamic Systems. In *Proceedings of 16th International Workshop on Principles of Diagnosis (DX 05)*.

[Reiter, 1992] Reiter, R. (1992). A Theory of Diagnosis From First Principles. In *Readings in Model-based Diagnosis*, pages 29–48, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Schwall and Gerdes, 2002] Schwall, M. and Gerdes, C. (2002). A probabilistic Approach to Residual Processing for Vehicle Fault Detection. In *Proceedings of the 2002 ACC*, pages 2552–2557.

[Stamatis, 1995] Stamatis, D. H. (1995). *Failure Mode and Effect Analysis: FMEA from Theory to Execution*. ASQ Quality Press.

[Vemuri and Polycarpou, 1997] Vemuri, A. T. and Polycarpou, M. M. (1997). Neural-network-based robust fault diagnosis in robotic systems. *IEEE Transactions on Neural Networks*, 8(6):1410–1420.

[Verron et al., 2007] Verron, S., Tiplica, T., and Kobi, A. (2007). Fault Diagnosis of Industrial Systems with Bayesian Networks and Mutual Information. In *Proceedings of the European Control Conference (ECC 07)*, pages 2304–2311.

[Warnquist et al., 2009] Warnquist, H., Pernestål, A., and Nyberg, M. (2009). Anytime near-optimal troubleshooting applied to an auxiliary truck braking system. In *Proceedings of 6th IFAC Symposium on Fault Detection, Supervision and Safety of technical processes (SAFEPROCESS 2009)*.

[Zhao et al., 2005] Zhao, F., Koutsoukos, X., Haussecker, H., Reich, J., and Cheung, P. (2005). Monitoring and fault diagnosis of hybrid systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 35(6):1225–1240.

Paper 2

Bayesian Inference by Combining Training Data and Background Knowledge Expressed as Likelihood Constraints[1]

Anna Pernestål and Mattias Nyberg

Division of Vehicular Systems, Department of Electrical Engineering,
Linköping University,
Sweden.

Abstract

Bayesian inference, or classification, from data is a powerful method for determining states of process when no detailed physical model of the process exists. However, the performance of Bayesian inference from data is dependent on the amount of training data available. In many real applications the amount of training data is limited, and inference results become insufficient. Thus it is important to take other kinds of information into account in the inference as well. In this paper, we consider a general type of background knowledge that appears in many real applications, for example medical diagnosis, technical diagnosis, and econometrics. We show how it can be expressed as constraints on the likelihoods, and provide detailed description of the computations. The method is applied to a diagnosis example, where it is clearly shown how the integration of background knowledge improves diagnosis when training data is limited.

[1]This paper has been submitted to International Journal of Approximate Reasoning. An earlier and shorter version is published as [Pernestål and Nyberg, 2007].

1 Introduction

We consider Bayesian inference, or classification, from both training data *and* background knowledge. The task is to compute the probability

$$p(C = c_l | \mathbf{X} = \mathbf{x}_k, \mathcal{D}, \mathbf{i}). \tag{1}$$

for the classes c_l, given an observation vector $\mathbf{x}_k$, training data $\mathcal{D}$, and background knowledge $\mathbf{i}$. If there is a large amount of training data available, sufficient classification results can often be obtained from training data only, without considering the background knowledge explicitly. In this case, methods based solely on training data can be used [Devroye et al., 1996, Heckerman et al., 1995, Sivia, 1996, Kontkanen et al., 2001]. However, in many real applications, for example fault diagnosis of technical processes, the amount of available training data is limited and the background knowledge $\mathbf{i}$ must be considered explicitly to achieve sufficient classification results. In fault diagnosis applications it is typically difficult to collect data from faulty situations, since faults are usually rare. One possibility is to actively implement faults and collect data. Still, faults may have severe or dangerous consequences which complicate data collection and limits the amount of training data available. Furthermore, the active implementation of faults in training data force us to handle experimental training data, i.e. training data with a different distribution than in the intended application.

We consider two sources of background knowledge. The first describes relations between likelihoods, i.e. the distributions of elements in the observation vector under certain assignments of the class variable, and can be found in a wide range of applications. In the diagnosis application this background knowledge indicates expected behavior of observations under certain faults. The same kind background knowledge can be found for example in medical and econometric applications [Niculescu et al., 2006, Giffin and Caticha, 2007, Feelders and van der Gaag, 2005]. The second source of background knowledge is the prior probability distribution of faults. This distribution is important knowledge, since the experimental training data does not tell anything about the distribution of faults.

The main contribution of the paper is a method for doing Bayesian inference, or classification, under the presence of both training data and background knowledge. To do this, we show how the background knowledge can be translated to likelihood constraints in the learning phase. We present the computations in detail, and also provide numerical methods for solving the integrals that appear. Finally we present two examples; the first illustrates and compares the analytical and the numerical solution methods, while the second show how the inference method can be applied to the task of diagnosis.

Bayesian inference based on training data alone is previously studied in for example [Devroye et al., 1996, Heckerman et al., 1995, Kontkanen et al., 2001,

Sivia, 1996]. However, Bayesian inference from both data and prior knowledge is previously only rarely discussed in literature. In [Feelders and van der Gaag, 2005, Niculescu et al., 2006] similar kinds of background knowledge as in the current paper are considered. These works focus on methods for modeling in Bayesian networks, while we focus directly on computing the probabilities, i.e. on doing inference. Furthermore, the kinds of background knowledge studied in these previous works are special cases of the more general type of background knowledge that we handle here. In [Pernestål and Nyberg, 2008] a diagnosis application is presented in which data and background knowledge are combined for inference. However, the background knowledge considered in the current paper is more general.

We begin by introducing notation and motivating the type of background knowledge we consider in Section 2. Background theory on Bayesian classification from data only is summarized in Section 3. We then show how background knowledge about the process can be expressed as likelihood constraints and extend the methods to also take these constraints into account in Section 4. We spend Section 5 on discussing numerical methods for solving the inference problem, and Section 6 we illustrate the effect of combining data and background knowledge in Bayesian inference in two examples. Finally, we discuss related work in Section 7 and conclude in Section 8.

2 Preliminaries

Before going into the computational details for determining the posterior probability (1) we introduce notation and discuss the kind of background knowledge considered.

2.1 Notation

We use the notation $p(Y = y|Z = z)$, or simply $p(y|z)$, for both probabilities and discrete probability distributions. For continuous probability density functions we write $f(y)$.

The inference problem is the task of determining the state, or rather the probability distribution of the state, of a process. The state of the process is described by a discrete scalar class variable C with domain $\mathbb{C} = \{c_1, \ldots, c_L\}$. A value c_l is referred to as a *class*. The process is observed by using a vector $\mathbf{X} = (X_1, \ldots, X_R)$. Element X_r in the observation vector has domain $\mathbb{X}_r = \{x_{r1}, \ldots, x_{rK_r}\}$, and the observation vector $\mathbf{X}$ has domain $\mathbb{X} = \mathbb{X}_1 \times \mathbb{X}_2 \times \ldots \times \mathbb{X}_R$. To denote an assignment of the complete observation vector we write $\mathbf{X} = \mathbf{x}_k$, $k = 1, \ldots, K$, where $K = \prod_{r=1}^{R} K_r$. Each value $\mathbf{x}_k$ is an R-dimensional vector, and we write $\mathbf{x}_k = (\mathbf{x}_k[1], \ldots, \mathbf{x}_k[R])$ to denote the elements explicitly. With this notation, $\mathbf{x}_k[r]$ is the value of X_r when $\mathbf{X} = \mathbf{x}_k$.

Training data samples d_i, $i = 1, \ldots, N$ consists of simultaneous values of the class variable C and the observation vector $\mathbf{X}$. A realization of training data is denoted $\mathcal{D}$.

We assume that the underlying process is such that the observation vectors are independent given the class. This assumption is called the Markov assumption, and is reasonable (at least approximately) in most real processes. In particular, in most real processes, data collection can be arranged such that this assumption is fulfilled. A more detailed discussion can be found in [Pernestål and Nyberg, 2008].

2.2 Background Knowledge

To motivate the type of background knowledge studied, let us consider the problem of fault diagnosis of technical processes. In diagnosis, the task is to compute the probabilities that different fault states (classes) c_l are present, given an observation vector $\mathbf{x}_k$ from the process. One approach is to apply traditional data driven methods for Bayesian inference as the ones presented in e.g. [Devroye et al., 1996, Heckerman et al., 1995, Kontkanen et al., 2001, Sivia, 1996] to compute the probabilities for faults. However, in these methods, it is crucial that there is training data available from all different faults (classes). In diagnosis applications there is typically training data from the fault free case and possibly from a limited set of faults. The reason is that faults often occur only rarely. One possibility to gain data from faulty cases is to actively implement faults and collect data. However, this is often an expensive or even dangerous approach, and therefore data can often only be collected from a subset of faults. Furthermore, implementing faults gives experimental data, i.e. the distribution of data in the training set is not the same as the distribution in the application. Due to the limitation in amount of training data, and the experimental characteristics of it, the traditional methods are often not sufficient for diagnosis applications.

On the other hand, there is often background knowledge available. The background knowledge consists of two parts: knowledge about the distribution of faults, and knowledge about which faults that may affect the different observations. In particular, it may be known that the effect on a certain observation is the same under two different faults. To see this, consider the following example.

Example 2.4 (Diagnosis).
In a process, there are three redundant sensors measuring the same temperature, see Figure 1. The sensor signals are denoted

$$T_i = T + \nu_i, \;\; i = 1, 2, 3,$$

where T is the true temperature and ν_i is a measurement noise. To monitor the

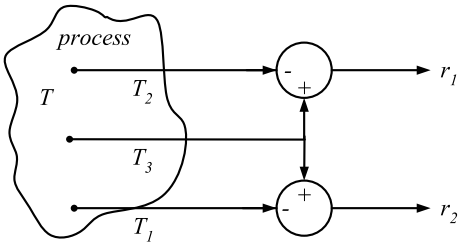


Figure 1: A sample process with temperature T measured by sensors giving temperature signals, $T_i, i = 1, 2, 3$. The sensor signals are used to form two residuals, $r_i, i = 1, 2$.

three sensors and detect faults in them two residuals are formed, $r_1 = T_3 - T_2$ and $r_2 = T_3 - T_1$. The residuals are discretized and used as observations X_1 and X_2 respectively.
We consider three classes: faults in sensors 1 and 2, and the fault free case. Rather than using notation c_i, $i = 1, 2, 3$, we use the more explaining notation F_1, F_2, and NF for the three classes. In this example X_1 has the same distribution under the two classes NF and F_1, while X_2 has the same distribution under the two classes NF and F_2. This information is our background knowledge, and we denote it **i**. The background knowledge can be compactly represented by the structure

	NF	F_1	F_2
X_1	1	1	2
X_2	3	4	3

(2)

In (2) the same number in two different columns means that the observation in the corresponding row is equally distributed under the two classes represented by the columns. The distribution of X_1 (or X_2) given C is unknown, but the background knowledge states that for some values of C the distribution of X_1 (or X_2) is the same.

The structure in (2) is an extension of a so called Fault Signature Matrix (FSM) or Fault Information System (FIS), see for example [Daigle et al., 2006, Korbicz et al., 2004, Pulido et al., 2005, Nyberg, 2002]. Since there are several slightly different interpretations of FSM/FIS it is difficult describe the exact relation between the background knowledge used in the current paper and FSM/FIS. However, the main idea with the FSM/FIS is to represent which observations that are affected by each faults (and possibly also in what direction), but no detailed information is given about whether the distribution of the observations changes. The structure (2) can easily be translated to an FSM by replacing figures appearing in the NF column with 0 (meaning "not affected"),

and all other figures with $\mathcal{X}$ (meaning "possibly affected"). In the structure (2), we change figures 1 and 3 to 0, and 2 and 4 to $\mathcal{X}$. This gives the FSM

	NF	F_1	F_2
X_1	0	0	$\mathcal{X}$
X_2	0	$\mathcal{X}$	0

In previous work on diagnosis using the FSM, there are two main approaches. The first approach is to apply logic based methods as in [de Kleer and Williams, 1992, Reiter, 1992], and the numerous works building on these references. However, in these approaches, the probabilistic information is not fully utilized, and diagnosis may be improved. The second approach is to use the FSM to introduce independence relations between observations. As shown in [Pernestål et al., 2006] this may lead to erroneous diagnosis results since observations not really are independent, but rather independent *sometimes*. In the current work we aim at utilizing all probabilistic information present in the background knowledge, but avoid introducing erroneous independence assumptions.

The kind of background knowledge as considered in the current work appear in a wide range of application areas. In [Giffin, 2007] econometric problems are shown to include similar type of knowledge, but formulated in different terms to suit the Maximum Entropy methods utilized. In [Feelders and van der Gaag, 2005] the same type of knowledge arise from medical doctors' expertise in diagnosis, and in [Niculescu et al., 2006] are several examples presented. However, their scope is to represent the information in models while we focus on computing the probabilities directly from given data and background knowledge. Furthermore, the method presented here handles even more general kinds of background knowledge than the previous works. Relations to previous work is discussed further in Section 7.

Return again to the structure (2). Without knowledge of this structure, we are forced to learn the distribution of $\mathbf{X} = (X_1, X_2)$ under F_1 only from the (limited) data from that fault case. However, the structure (2) indicates that observation X_1 has the same distribution under the classes NF and F_1. This means that it is possible to reuse data from NF to learn about X_1 also under class F_1, and thus data from NF gives information also about the distribution of $\mathbf{X}$ under the class F_1. In the following sections we show how this reuse of data can be done formally.

3 Inference Using Data Only

First, we consider computation of the posterior probabilities $p(c_l|\mathbf{x}_k, \mathcal{D})$ when no background knowledge is given and computations must rely on training data only. We use the methods described for example in [Heckerman et al., 1995, Pernestål and Nyberg, 2008], but in order to be prepared for adding background

knowledge we rewrite the problem by using the product rule of probabilities to obtain

$$p(c_l|\mathbf{x}_k, \mathcal{D}) = \frac{p(c_l, \mathbf{x}_k|\mathcal{D})}{p(\mathbf{x}_k|\mathcal{D})}.$$

The denominator $p(\mathbf{x}_k|\mathcal{D}) = \sum_{l=1}^{L} p(c_l, \mathbf{x}_k|\mathcal{D})$ is independent of C and is thus constant for a given value $\mathbf{x}_k$ of the observation vector. Thus, given $\mathbf{x}_k$, the posterior probability for the class is proportional to the joint probability of C and $\mathbf{X}$, i.e.

$$p(c_l|\mathbf{x}_k, \mathcal{D}) \propto p(c_l, \mathbf{x}_k|\mathcal{D}) = p(\mathbf{z}_{lk}|\mathcal{D}), \tag{3}$$

where we have introduced the variable $\mathbf{Z} = (C, \mathbf{X})$. Let $\mathbf{Z}$ take values $\mathbf{z}_{lk} = (c_l, \mathbf{x}_k)$, and have domain $\mathbb{Z} = \mathbb{C} \times \mathbb{X}$. Each value $\mathbf{z}_{lk}$ is a vector that can take $M = KL$ different values. Sometimes it is more convenient to enumerate the values of $\mathbf{Z}$ as $\mathbf{z}_1, \ldots, \mathbf{z}_M$. There is a unique transformation from the double subscript $\mathbf{z}_{lk}$ to the single subscript $\mathbf{z}_m$. However, the exact representation of this transformation is not important and will not be discussed explicitly.

The computations of (3) are given in detail in for example [Heckerman et al., 1995, Pernestål and Nyberg, 2008]. Here we are content to summarize them in the following theorem.

Theorem 1. *Let $\mathbf{Z}$ be a discrete variable with $1, \ldots, M$ possible values. Introduce parameters $\mathbf{\Theta} = (\Theta_1, \ldots, \Theta_M)^T$ with values $\theta = (\theta_1, \ldots, \theta_M)^T$ such that*

$$p(\mathbf{z}_m|\theta) = \theta_m, \quad m = 1, \ldots, M, \tag{4a}$$

$$\theta_m > 0 \tag{4b}$$

$$\sum_{m=1}^{M} \theta_m = 1. \tag{4c}$$

Let $f_{\mathbf{\Theta}}(\theta)$ be Dirichlet distributed[2], i.e.

$$f_{\mathbf{\Theta}}(\theta) = \frac{\Gamma(\sum_{m=1}^{M} \alpha_m)}{\prod_{m=1}^{M} \Gamma(\alpha_m)} \prod_{m=1}^{M} \theta_m^{\alpha_m - 1}, \quad \alpha_m > 0, \tag{5}$$

where $\Gamma(\cdot)$ is the gamma function, and the parameters $\alpha = (\alpha_1, \ldots, \alpha_M)$ are given. Let $\mathcal{D}$ be a (possibly empty) set independent samples of $\mathbf{Z}$. Let n_m be the count of samples in $\mathcal{D}$ where $\mathbf{Z} = \mathbf{z}_m$, and let $N = \sum_{m=1}^{M} n_m$ and $A = \sum_{m=1}^{M} \alpha_m$. Then it holds that

$$p(\mathbf{z}_m|\mathcal{D}) = \frac{n_m + \alpha_m}{N + A}. \tag{6}$$

[2]In fact, it can be shown that under regular assumptions, the Dirichlet distribution on $\mathbf{\Theta}$ is inevitable [Geiger and Heckerman, 1997]

Note that there is a relation between the parameters α_m in (6) and the prior probabilities $p(c_l)$ for the classes. As noted in the paragraph below equation (3) there is a unique transformation between the single subscript m and the double subscript lk. Using the same transformation we can write either α_m or α_{lk}. To avoid clutter we use α_{lk} to express the relation between the parameters and the prior probabilities for the classes as

$$p(c_l) = \sum_{k=1}^{K} p(c_l, \mathbf{x}_k) = \frac{1}{A} \sum_{k=1}^{K} \alpha_{lk}, \tag{7}$$

meaning that the prior probabilities for the classes are sums of the parameters α_{lk} (or, equivalently α_m).

4 Inference Using Data and Background Knowledge

Now, assume that in addition to the training data we are also given knowledge of the type presented in Section 2.2. In this section, we first show how this type of knowledge is represented as constraints on the parameters $\mathbf{\Theta}$. We then derive expressions for computing the probability for $\mathbf{Z}$ given both data and constraints.

4.1 Background Knowledge as Constraints

In Section 2.2 we introduced the type of background knowledge considered in the paper. Here we will formalize the background knowledge as a random variable. The background knowledge includes two parts of information:

1. It specifies that the data generating process is such that there are elements in the observation vector that are equally distributed under different classes.

2. It specifies the probability for the classes.

We let the background knowledge be represented by the random variable $\mathbf{I}$ which take value $\mathbf{i}$. Part (i) means that for a given background knowledge $\mathbf{i}$ it is specified exactly under which classes the observations are equally distributed. This part of the background knowledge is represented by a set $\mathcal{E}$ of tuples of the type

$$\begin{aligned} &\langle r_1, r_2, k_1, k_2, l_1, l_2 \rangle \,, \\ &r_1, r_2 \in \{1, \ldots, R\}, k_1 \in \{1, \ldots, K_{r_1}\}, k_2 \in \{1, \ldots, K_{r_2}\}, l_1, l_2 \in \{1, \ldots, L\}. \end{aligned} \tag{8}$$

The tuple in (8) represents the statement "the parameters $\mathbf{\Theta}$ are such that the relation $p(x_{r_1 k_1}|c_{l_1}, \theta) = p(x_{r_2 k_2}|c_{l_2}, \theta)$ holds".

Part (ii) of the background information states that "the parameters $\mathbf{\Theta}$ are such that

$$\sum_{\mathbf{x}\in\mathbb{X}} p(c_l, \mathbf{x}|\theta) = p(c_l|\theta) = p_{c_l}, \tag{9}$$

holds". Part (ii) of the background knowledge is represented by a vector $\mathcal{P} = (p_{c_1}, \ldots, p_{c_L})$. In particular, note that part (ii) of the background knowledge implies that $p(c_l|\mathcal{D}, \mathbf{i}) = p(c_l|\mathbf{i})$

The sample space of the random variable $\mathbf{I}$ is thus the set of all possible sets of tuples $\langle \mathcal{E}, \mathcal{P} \rangle$[3].

From the discussion above it follows that given the background knowledge it can be stated that constraints of the type

$$p(x_{r_1 k_1}|c_{l_1}, \theta, \mathbf{i}) = p(x_{r_2 k_2}|c_{l_2}, \theta, \mathbf{i}) \tag{10}$$

hold. To simplify the notation in the computations below, we restrict (10) to the case where $r_1 = r_2 = r$, $k_1 = k_2 = k$, and $l_1 = j$, $l_2 = l$. Then (10) becomes

$$p(x_{rk}|c_j, \theta, \mathbf{i}) = p(x_{rk}|c_l, \theta, \mathbf{i}). \tag{11}$$

Applying the product rule of probabilities on (11) we have

$$\frac{p(c_j, x_{rk}|\theta, \mathbf{i})}{p(c_j|\theta, \mathbf{i})} = p(x_{rk}|c_j, \theta, \mathbf{i}) = p(x_{rk}|c_l, \theta, \mathbf{i}) = \frac{p(c_l, x_{rk}|\theta, \mathbf{i})}{p(c_l|\theta, \mathbf{i})},$$

where $p(c_j|\theta, \mathbf{i}) = p_{c_j}$ and $p(c_l|\theta, \mathbf{i}) = p_{c_l}$ according to (9). The prior probabilities can be written $pc_j = \rho_{jl} pc_l$ with a known constant ρ_{jl}. Thus, (11) means that

$$p(c_j, x_{rk}|\theta, \mathbf{i}) = \rho_{jl} p(c_l, x_{rk}|\theta, \mathbf{i}). \tag{12}$$

To relate the distributions in (12) to distributions of $\mathbf{Z}$, we marginalize over all possible values of the elements in $\mathbf{X}$ except X_r. Let

$$\mathbf{X}_{\bar{r}} = (X_1, \ldots, X_{r-1}, X_{r+1}, \ldots, X_R),$$

let $\mathbf{x}_{\bar{r}}$ denote an assignment of $\mathbf{X}_{\bar{r}}$, and let $\mathbb{X}_{\bar{r}}$ be the domain of $\mathbf{X}_{\bar{r}}$. Then we can write

$$p(c_j, X_r = x_{rk}|\theta, \mathbf{i}) = \sum_{\mathbf{x}_{\bar{r}}\in\mathbb{X}_{\bar{r}}} p(c_j, X_r = x_{rk}, \mathbf{X}_{\bar{r}} = \mathbf{x}_{\bar{r}}|\theta, \mathbf{i}) =$$

$$= \sum_{\mathbf{z}_m \in \mathbb{Z}_{x_{rk}, c_j}} p(\mathbf{z}_m|\theta, \mathbf{i}) = \sum_{\mathbf{z}_m \in \mathbb{Z}^m_{x_{rk}, c_j}} \theta_m \tag{13}$$

[3]The random variable $\mathbf{I}$ has a probability distribution $p(\mathbf{I} = \mathbf{i})$. Since $\mathbf{I}$ always appears on the right hand side of the |-sign in the probabilities there is no need for expressing $p(\mathbf{I} = \mathbf{i})$ explicitly. However, as a comment, it is indeed possible to talk about the probability that certain background knowledge is present, since it is related to the structure of the processes.

where $\mathbb{Z}_{x_{rk},c_j} = \{\mathbf{z}_m \in \mathbb{Z} : \mathbf{z}_m = (c_j, \mathbf{x}_q), \mathbf{x}_q[r] = x_{rk}\}$, i.e. the set of all possible values $\mathbf{z}_m$ where $C = c_j$ and $X_r = x_{rk}$, regardless of the values of the other elements in the observation vector. By using (13) we can write the requirement (12) in the form

$$\sum_{\substack{m \\ \mathbf{z}_m \in \mathbb{Z}_{x_{rk},c_j}}} \theta_m = \rho_{jl} \sum_{\substack{m \\ \mathbf{z}_m \in \mathbb{Z}_{x_{rk},c_j}}} \theta_m. \tag{14}$$

The part of the background knowledge concerning the probabilities, given by (9) can be expressed as constraints on θ as follows.

$$p_{c_l} = \sum_{\mathbf{x} \in \mathbb{X}} p(c_l, \mathbf{x}|\theta) = \sum_{\substack{m \\ \mathbf{z}_m \in \mathbb{Z}_{c_j}}} \theta_m. \tag{15}$$

where $\mathbb{Z}_{c_j} = \{\mathbf{z}_m \in \mathbb{Z} : \mathbf{z}_m = (c_j, \mathbf{x}), \mathbf{x} \in \mathbb{X}\}$, i.e. the set of all possible values $\mathbf{z}_m$ where $C = c_j$ regardless of the value of $\mathbf{X}$. We illustrate how the background knowledge can be expressed as parameter constraints with the following example.

Example 4.5 (Two Classes).
Consider the case with two classes, $C \in \{c_1, c_2\}$, and a one-dimensional observation $\mathbf{X} \in \{\mathbf{x}_1, \mathbf{x}_2\}$. Define $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$ by

$$p(c_1, \mathbf{x}_1|\theta, \mathbf{i}) = \theta_1,\ p(c_2, \mathbf{x}_1|\theta, \mathbf{i}) = \theta_2,$$
$$p(c_1, \mathbf{x}_2|\theta, \mathbf{i}) = \theta_3,\ p(c_2, \mathbf{x}_2|\theta, \mathbf{i}) = \theta_4.$$

Assume that we are given the background knowledge that $p(\mathbf{x}_1|c_1, \theta) = p(\mathbf{x}_1|c_2, \theta)$ and that both classes are equally probable, i.e. that $p_{c_1} = p_{c_2} = 0.5$. Expressed in terms of the parameters this means that

$$\theta_1 = \theta_2,$$
$$\theta_1 + \theta_3 = \theta_2 + \theta_4 = 0.5.$$

Constraints of the forms (14) and (15) can be written on the general form

$$F\boldsymbol{\Theta} = G, \tag{16}$$

where $F \in \mathbb{R}^{S \times M}$ and $G \in \mathbb{R}^S$. Several types of background knowledge, including (10), can be represented in the form (16). As will be illustrated in the example in Section 6.2, this is typical in diagnosis applications. The constraints we consider here are more general than previous works such as [Boutilier et al., 1996] and [Jaeger et al., 2005], where constraints on single parameters are studied. These previous works are special cases of the constraints we study here. The relation to these previous works is further considered in Section 7.

The number S of rows in F and G, is the number of constraints, including (10) and (9) Here follows an example of how F and G may look like.

Example 4.6 (Example 4.5 cont.).
With the parameters in Example 4.5, and with $\rho_{12} = 1$, the matrices in (16) becomes

$$F = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix}. \tag{17}$$

Note that the two last rows in F guarantees that our parameters θ sum to 1, i.e. that constraint(4c) is fulfilled.

4.2 Computing the Probability of Z under constraints

Now, return to the computation of $p(\mathbf{z}_m|\mathcal{D}, \mathbf{i})$. There are M possible values of $\mathbf{Z}$ meaning that there are M parameters $\boldsymbol{\Theta} = (\Theta_1, \ldots, \Theta_M)$ needed to describe the distribution $p(\mathbf{Z}|\mathcal{D}, \mathbf{i})$. The requirement (16) decreases the degree of freedom, and the parameters $\boldsymbol{\Theta}$ can be expressed by $Q = M - S$ parameters $\boldsymbol{\Phi} = (\Phi_1, \ldots, \Phi_Q)$. Use $\phi = (\phi_1, \ldots, \phi_Q)$ to denote the values of $\boldsymbol{\Phi}$. There is a linear transformation

$$\boldsymbol{\Theta} = V\boldsymbol{\Phi} + U \tag{18}$$

between the $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$. The transformation matrices will be derived in Section 4.3. By know, we simply assume that they exist.

Let $\Delta_{\boldsymbol{\Phi}}$ be the set of parameters $\boldsymbol{\Phi}$ such that their transformation (18) fulfills (4b). Marginalizing gives

$$p(\mathbf{z}_m|\mathcal{D}, \mathbf{i}) = \int_{\Delta_{\boldsymbol{\Phi}}} p(\mathbf{z}_m|\phi, \mathcal{D}, \mathbf{i}) f(\phi|\mathcal{D}, \mathbf{i}) d\phi. \tag{19}$$

For the first factor in the integrand we note that when the parameters ϕ are known, then $\mathbf{Z}$ is independent of $\mathcal{D}$[4]. Thus, we have

$$p(\mathbf{z}_m|\phi, \mathcal{D}, \mathbf{i}) = p(\mathbf{z}_m|\phi, \mathbf{i}) = p(\mathbf{z}_m|\theta, \mathbf{i}) = \theta_m, \tag{20}$$

where we have used (16) in the second equality. To determine the second factor in the integrand of (19), apply Bayes' theorem to obtain

$$f(\phi|\mathcal{D}, \mathbf{i}) = \frac{p(\mathcal{D}|\phi, \mathbf{i}) f_{\boldsymbol{\Phi}}(\phi|\mathbf{i})}{\int_{\Delta_{\boldsymbol{\Phi}}} p(\mathcal{D}|\phi, \mathbf{i}) f_{\boldsymbol{\Phi}}(\phi|\mathbf{i}) d\phi}. \tag{21}$$

[4] for details, see [Pernestål and Nyberg, 2008].

Let training sample number i taken the value $\mathbf{z}_{\mu_i}$. Then, by using the fact that training samples are independent, we can rewrite the factor $p(\mathcal{D}|\phi, \mathbf{i})$ in (21) as

$$p(\mathcal{D}|\phi, \mathbf{i}) = \prod_{i=1}^{N} p(\mathbf{z}_{\mu_i}|\phi, \mathbf{i}) = \prod_{i=1}^{N} p(\mathbf{z}_{\mu_i}|\theta, \mathbf{i}) = \theta_1^{n_1} \dots \theta_M^{n_M}, \tag{22}$$

where n_m is the number of samples in training data where $\mathbf{Z} = \mathbf{z}_m$, and $\sum_{i=m}^{M} n_m = N$. Let V_m and U_m be the m:th rows in V and U respectively. Then $\theta_m = V_m\phi + U_m$, and (22) becomes

$$p(\mathcal{D}|\phi, \mathbf{i}) = (V_1\phi + U_1)^{n_1} \dots (V_M\phi + U_M)^{n_M}. \tag{23}$$

For the second factor of (21), the prior probability for $\mathbf{\Phi}$ we use

$$f_{\mathbf{\Phi}}(\phi|\mathbf{i}) = \begin{cases} \gamma f_{\mathbf{\Theta}}(V\phi + U|\mathbf{i}) & \text{if } V\phi + U > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{24}$$

where $f_{\mathbf{\Theta}}$ is defined by (5) and the requirement $V\phi + U > 0$ comes from (4b). In (24), γ is a constant that guarantees that $f_{\mathbf{\Phi}}$ integrates to one. Intuitively (24) means that the background knowledge about the constraints simply cut off the parts of $f_{\mathbf{\Theta}}$ that is not consistent with $\mathbf{i}$ and rescale the distribution in a space with smaller dimension while the "shape" is the same as before introducing the constraints.

By using equations (4), (19), (20), (21), (22), and (24) we obtain

$$p(\mathbf{Z} = \mathbf{z}_m|\mathcal{D}, \mathbf{i}) = \\ = \frac{\int_{\Delta_{\mathbf{\Phi}}} (V_1\phi + U_1)^{n_1+\alpha_1-1} \dots (V_m\phi + U_m)^{n_m+\alpha_m} \dots (V_M\phi + U_M)^{n_M+\alpha_M-1} d\phi}{\int_{\Delta_{\mathbf{\Phi}}} (V_1\phi + U_1)^{n_1+\alpha_1-1} \dots (V_m\phi + U_m)^{n_m+\alpha_m-1} \dots (V_M\phi + U_M)^{n_M+\alpha_M-1} d\phi}. \tag{25}$$

4.3 Parameter Transformation

To solve the integrals in (25) we will now derive the explicit relation between $\mathbf{\Theta}$ and $\mathbf{\Phi}$, and then also an expression for the region $\Delta_{\mathbf{\Phi}}$ of integration.

First, note that the matrix $F \in \mathbb{R}^{S\times M}$ has full row rank, otherwise there would be redundant information about the parameters θ, and rows could be removed from F. Thus, we can always order the constraints, for example by multiplying (16) with a permutation matrix, so that $F = [F_S \quad F_{M-S}]$ where $F_S \in \mathbb{R}^{S\times S}$ has full rank.

The constraints (16) can then be rewritten as

$$\begin{bmatrix} I & F_S^{-1}F_{M-S} \end{bmatrix} \mathbf{\Theta} = F_S^{-1}G. \tag{26}$$

Let $A_{i:j}$ denote rows i to j in the matrix A. We can then write (26) as

$$\mathbf{\Theta}_{1:S} + F_S^{-1}F_{M-S}\mathbf{\Theta}_{S+1:M} = F_l^{-1}G. \tag{27}$$

Letting $\mathbf{\Phi} = \mathbf{\Theta}_{S+1:M}$ and rearranging the terms of (27) gives

$$\mathbf{\Theta} = \underbrace{\begin{bmatrix} -F_S^{-1}F_{M-S} \\ I \end{bmatrix}}_{V} \mathbf{\Phi} + \underbrace{\begin{bmatrix} F_S^{-1}G \\ 0 \end{bmatrix}}_{U}.$$

The region $\Delta_{\mathbf{\Phi}}$ of integration is determined by for each Φ_i, $i = 1, \ldots, Q$ one optimization problem for the lower boundary and one for the upper boundary. For the lower boundary it is given by

$$\begin{aligned} &\phi_i^L = \min \phi_i \\ &\text{subject to} \quad V\Phi + U > 0 \\ &\qquad\qquad \phi_j > \phi_j^L \quad j = 1, \ldots, i-1. \end{aligned} \tag{28}$$

For the upper boundary the optimization problem is given by

$$\begin{aligned} &\phi_i^U = \max \phi_i \\ &\text{subject to} \quad V\Phi + U > 0 \\ &\qquad\qquad \phi_j < \phi_j^U \quad j = 1, \ldots, i-1. \end{aligned} \tag{29}$$

To investigate the computations in detail, consider the following example.

Example 4.7 (Example 4.6 cont.).
For Examples 4.5 and 4.6 the matrices F and G are given by (17). This gives $V = [-1 \ -1 \ 1 \ 1]^T$ and $U = [0.5 \ 0.5 \ 0 \ 0]^T$, i.e. we have a scalar $\mathbf{\Phi} = \Phi_1$ and the relations $\Theta_1 = 0.5 - \mathbf{\Phi}_1$, $\Theta_2 = 0.5 - \mathbf{\Phi}_1$, $\mathbf{\Theta}_3 = \mathbf{\Phi}_1$, and $\mathbf{\Theta}_4 = \mathbf{\Phi}_1$. Solving the optimization problems (28) and (29) we obtain $\Phi_1^L = 0$ and $\Phi_1^U = 0.5$. Thus, the integrals in (25) becomes

$$\begin{aligned} &\int_0^{0.5} (0.5 - \phi_1)^{k_1} (0.5 - \phi_1)^{k_2} \phi_1^{k_3} \phi_1^{k_4} d\phi_1 = \\ &= \frac{1}{2^{1+\sum_{i=1}^4 k_i}} \frac{\Gamma(k_1 + k_2 + 1)\Gamma(k_3 + k_4 + 1)}{\Gamma(2 + \sum_{i=1}^4 k_i)}, \end{aligned}$$

where $k_i = n_i + \alpha_i - 1$ or $k_i = n_i + \alpha_i$ depending on the value of i and whether we solve the integral in the denominator or numerator of (25).

5 Computing the Integrals

The integrals in (25) are of the type

$$\int_{\Delta_{\mathbf{\Phi}}} (V_1\phi + U_1)^{k_1} \ldots (V_M\phi + U_M)^{k_M} d\phi, \tag{30}$$

for nonnegative integers k_m. Although an analytical solution was easily found in Example 4.7, this is generally not the case. To the authors knowledge, there is in general no closed form solution to (30). In this section we study how an approximation method can be used.

5.1 Characteristics of the Integral

Consider the integral (30), and denote the integrand $h(\phi) = \prod_{m=1}^{M}(V_m\phi + U_m)^{k_m}$. To solve the integral approximately, we use the following proposition.

Proposition 1 (Multivariate Unimodal)**.** *Let $k_m > 0$ for at least one m, then the function $h(\phi)$ is multivariate unimodal*[5] *inside $\Delta_{\mathbf{\Phi}}$.*

Proof. For $k_m > 0$, h is zero at the borders of $\Delta_{\mathbf{\Phi}}$: either we have $\phi_m = 0$ or $(V_i\phi + U_i) = 0$. Each factor in the integrand is positive inside $\Delta_{\mathbf{\Phi}}$, thus the integrand must have at least one maximum inside the region of integration.

To show that there is only one maximum we study the Hessian of $h_L(\phi) = \log h(\phi)$ inside $\Delta_{\mathbf{\Phi}}$. Let V_{ip} denote the p:th element in the row vector V_i. The first and second derivatives of h are then

$$\frac{\partial h_L}{\partial \phi_p} = \sum_{i=1}^{M} \frac{V_{ip}k_i}{V_i\phi + U_i},$$

$$\frac{\partial^2 h_L}{\partial \phi_p \partial \phi_q} = -\sum_{i=1}^{M} \frac{V_{ip}V_{iq}k_i}{(V_i\phi + U_i)^2},$$

and the Hessian of h_L can be written

$$H = \left[\nabla^2 h_L(\phi)\right] = \left[\frac{\partial^2 h_L}{\partial \phi_p \partial \phi_q}\right] = -\sum_{i=1}^{M} \frac{k_i}{(V_i\phi + U_i)^2} V_i^T V_i. \tag{31}$$

The Hessian is clearly negative semidefinite. To see that H is also negative definite, note that for a general vector x we have $x^T H x = -\sum_{i=1}^{M} 1/(V_i\phi + U_i)^2 (x \cdot V_i)^2$. Since Δ_ϕ is finite $(x \cdot V_i) \neq 0$ for at least one i and thus H is negative definite. □

Now, we use the fact that $h(\phi)$ is unimodal and approximate it with another unimodal function with known integral. The case with $k_m = 0$ for all $m = 1, \ldots, M$ corresponds to the case where no training data exists and gives a constant integrand and solving the integral is trivial.

One way to estimate the integral of a multivariate unimodal function that is small (zero) at the boundaries of the region of integration, is to approximate the

[5] For multivariate functions there are several similar definitions of "multivariate unimodal", see [Dharmadhikari and Joag-Dev, 2006]. Here, we mean that the function has exactly one maximum inside the region considered.

integrand by an unnormalized Gaussian distribution centered at the maximum value of the integrand. The integration of the unnormalized Gaussian can then be performed over the whole space, which provides an approximate solution. This approximation method is referred to as *Laplace approximation* [MacKay, 2005] or the *saddle point approximation* [Goutis and Casella, 1999].

Consider the problem of integrating $h(\phi)$ over the region $\Delta_{\mathbf{\Phi}}$. The Laplace approximation is given by

$$\int_{\Delta_{\mathbf{\Phi}}} h(\phi)d\phi \approx h(\phi^*) \int_{\mathbb{R}^Q} e^{-\frac{1}{2}(\phi-\phi^*)^T(-H(\phi^*))(\phi-\phi^*)}d\phi = h(\phi^*)\sqrt{\frac{(2\pi)^Q}{\det(-H(\phi^*))}}, \tag{32}$$

where

$$\phi^* = \arg\max_{\phi\in\Delta_{\mathbf{\Phi}}} h(\phi), \tag{33}$$

and $H(\phi^*)$ is given by (31). When performing the approximation in (32) the region of integration is changed from Δ to $\mathbb{R}^Q$, since $e^{-\frac{1}{2}(\phi-\phi^*)^T(-H)(\phi-\phi^*)}$ is approximately zero outside our region of integration. Without doing the approximation of the integrand this change of region is not applicable, since $h(\phi)$ is generally not zero (or even small) outside the region of integration.

The Laplace approximation utilizes the fact that the integrand is small on the boundaries. For the integrand of (30) to be small (zero) at the boundaries, it is required that $k_m > 0$ for all m, i.e. that there is training data available or that $\alpha_m > 1$. If $k_m = 0$ for some m the integrand becomes constant in some directions. In this case, integration is easily performed along these directions before the Laplace approximation is applied on the remaining, unimodal integrand.

The Laplace approximation is widely used in literature. However, to the authors' knowledge, there are no general boundaries on the errors in the approximation. Instead, each problem must be studied and the performance of the approximation verified, and in Section 6 we study its appropriateness to integrands of the type $h(\phi)$ used in the current paper.

6 Examples

We illustrate the computations of probabilities given data and constraints as presented in Sections 4 and 5 with two examples. The objective with the first, relatively small, example is to compare the Laplace approximation with the analytical solution. In the second, larger example we show how the method can be applied to a more realistic diagnosis problem.

6.1 Analytical Solution vs. Laplace Approximation

Consider the case where there are two classes c_1 and c_2, two binary observations, $\mathbf{X} = (X_1, X_2)$, $X_i \in \{0, 1\}$. The background knowledge $\mathbf{i}$ states that the two classes are equally probable, that all $\alpha_i = 1$, and that the parameters $\boldsymbol{\Theta}$ are such that

$$p(X_1 = x_{1k}|\theta, c_1) = p(X_1 = x_{1k}|\theta, c_2). \tag{34}$$

Enumerate the values of $\mathbf{Z}$ according to

C	1	2	1	2	1	2	1	2
X_1	0	0	1	1	0	0	1	1
X_2	0	0	0	0	1	1	1	1
	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	$\mathbf{z}_4$	$\mathbf{z}_5$	$\mathbf{z}_6$	$\mathbf{z}_7$	$\mathbf{z}_8$
	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7	Θ_8

(35)

The background knowledge expressed in the parameters is then

$$\begin{aligned}
\Theta_1 + \Theta_3 &= \Theta_2 + \Theta_6 \\
\Theta_3 + \Theta_7 &= \Theta_4 + \Theta_8 \\
\Theta_1 + \Theta_3 + \Theta_5 + \Theta_7 &= 0.5 \\
\Theta_2 + \Theta_4 + \Theta_6 + \Theta_8 &= 0.5
\end{aligned}$$

The last constraint above is a linear combination of the first three, and can be removed. Performing the variable transformation we obtain $\boldsymbol{\Phi} = V\boldsymbol{\Theta} + U$, where

$$V = \begin{bmatrix}
-1 & -1 & 0 & 0 & -1 \\
1 & 0 & -1 & 0 & -1 \\
1 & 0 & 0 & -1 & 1 \\
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}, \quad U = \begin{bmatrix}
0.5 \\ 0.5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix}. \tag{36}$$

The $Q = M - S$ last variables in $\boldsymbol{\Theta}$ form the new parameters, i.e.

$$\boldsymbol{\Phi} = (\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5)^T = (\Theta_4, \Theta_5, \Theta_6, \Theta_7, \Theta_8)^T.$$

To obtain the borders of integration we solve the optimization problems (28) and (28) for all ϕ_i. This gives the borders

$$\begin{aligned}
&0 < \phi_1 < 0.5,\\
&0 < \phi_2 < 0.5 - \phi_1 - \phi_5,\\
&0 < \phi_3 < 0.5 - \phi_1 - \phi_5,\\
&0 < \phi_4 < \phi_1 + \phi_5,\\
&0 < \phi_5 < 0.5 - \phi_1.
\end{aligned}$$

The integral (30) becomes

$$\begin{aligned}
&\int_{\Delta_\Phi} (0.5 - \phi_1 - \phi_2 - \phi_5)^{k_1} (0.5 - \phi_1 - \phi_3 - \phi_5)^{k_2} \times \dots\\
&(\phi_1 - \phi_4 + \phi_5)^{k_3} \phi_1^{k_4} \phi_2^{k_5} \phi_3^{k_6} \phi_4^{k_7} \phi_5^{k_8} d\Phi.
\end{aligned} \tag{37}$$

Let $\mathbf{n} = (n_1, \dots, n_8)$ be the vector of number of training samples when the values of $\mathbf{Z}$ are enumerated according to (35).

We exemplify the computations by considering the case where the system is set into class c_2 training data is collected. The result is a set of samples where n_4 and n_8 are the only nonzero elements in $\mathbf{n}$. To compare the analytical and approximate solutions, we compute the probability that $(C, X_1, X_2) = (c_1, 1, 0)$. From (35) we see that this value corresponds to $\mathbf{z}_3$ with probability Θ_3, which after the variable transformation is given by $\phi_1 - \phi_4 + \phi_5$. This gives the expression

$$\begin{aligned}
&p(C = c_1, X_1 = 1, X_2 = 0 | \mathcal{D}, \mathbf{i}) = p(\mathbf{z}_3 | \mathcal{D}, \mathbf{i}) =\\
&= \frac{\int_{\Omega_F} (\phi_1 - \phi_4 + \phi_5) \phi_1^{n_4} \phi_5^{n_8} d\phi_1 \dots d\phi_5}{\int_{\Omega_F} \phi_1^{n_4} \phi_5^{n_8} d\phi_1 \dots d\phi_5},
\end{aligned} \tag{38}$$

where the computations can be performed straight-forward.

To apply the Laplace approximation, consider the integrals (38). The integrand in the numerator is independent of ϕ_2 and ϕ_3, and linear in ϕ_4. The integrand in the denominator is independent of ϕ_2, ϕ_3, and ϕ_4. Therefore, both integrands are not unimodal in these directions, and we need to integrate analytically along them before applying the Laplace approximation. After the analytical integration we obtain

$$\frac{1}{2} \int_{\substack{\phi_1, \phi_5 > 0 \\ 0 < \phi_1, \phi_5 < 0.5}} (\phi_1 + \phi_5)^2 (0.5 - \phi_1 - \phi_5)^2 \phi_1^{n_4} \phi_5^{n_8} d\phi_1 d\phi_5 \tag{39a}$$

$$\int_{\substack{\phi_1, \phi_5 > 0 \\ 0 < \phi_1, \phi_5 < 0.5}} (\phi_1 + \phi_5) (0.5 - \phi_1 - \phi_5)^2 \phi_1^{n_4} \phi_5^{n_8} \phi_1 d\phi_5 \tag{39b}$$

for the numerator and denominator, respectively. These integrals are solved by using the Laplace Approximation.

Table 1: Analytical and Laplace Approximation Solutions of $p(\mathbf{z}_3|\mathcal{D}, \mathbf{i})$ for three sets of training data.

Data	Analytical Solution	Laplace Approximation
$n_4 = 4, n_8 = 2$	0.188	0.198
$n_4 = 10, n_8 = 5$	0.214	0.223
$n_4 = 40, n_8 = 20$	0.239	0.242

The results for three different sets of training data are summarized in Table 1. The error in the Laplace approximation is in most cases less than 5%, and decreases as the number of training data increases. In Figure 2 the integrand in (39a) and its Laplace approximation are plotted as functions of ϕ_1 and ϕ_5 when $n_4 = 10$ and $n_8 = 5$. In both plots the other variable is fixed at its value at ϕ^* defined by (33). The true (solid) curve and the approximated (dashed) curve are similar.

6.2 Diagnosis Example

Now, we apply the method to an extended version of the scenario presented in Example 2.4, where there are three sensors measuring the same temperature. The objective now is to detect and localize single and multiple faults in the sensors. The sensor signals are denoted T_1, T_2 and T_3. We construct three residuals,

$$\begin{aligned} r_1 &= T_3 - T_2, \\ r_2 &= T_3 - T_1, \\ r_3 &= T_2 - T_1. \end{aligned}$$

The observations are formed by discretizing the residuals in two bins, i.e. we form binary observations.

Let NF denote the fault free case, and F_i denote fault in sensor i. Considering single and double faults, we obtain the structure (40) that represents our background knowledge.

	NF	F_1	F_2	F_3	$F_1\&F_2$	$F_1\&F_3$	$F_2\&F_3$
X_1	0	0	1	2	1	2	3
X_2	4	5	4	6	5	7	6
X_3	8	9	10	8	11	9	10

(40)

In this structure, the same number means that the corresponding marginal distributions are the same, for example that $p(x_1|C = NF) = p(x_1|C = F_1)$. With three binary observations and seven classes we have $2^3 \cdot 7 = 56$ parameters

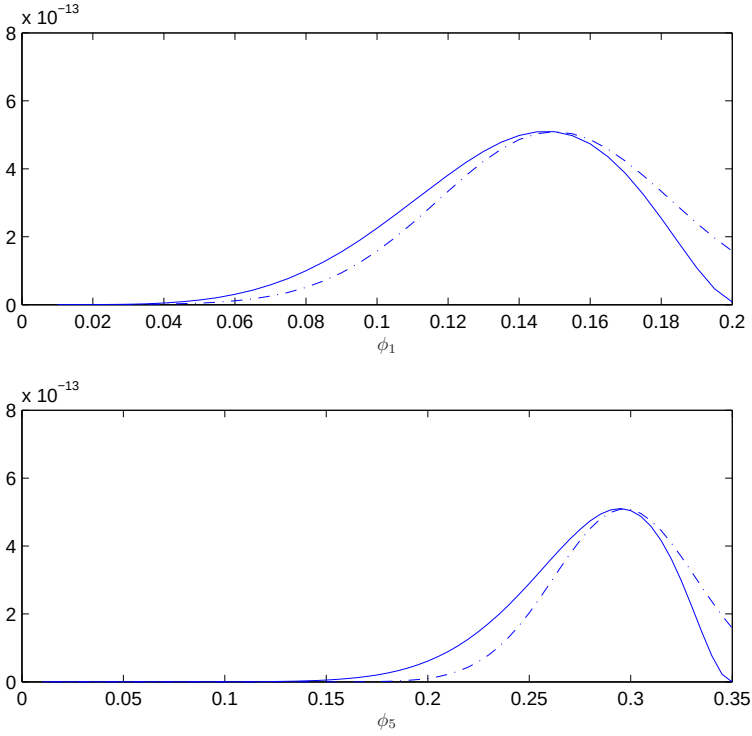


Figure 2: The integrand (solid) in (39a) and its Laplace approximation (dashed) as a function of ϕ_1 (top) and ϕ_5 (bottom). In both plots the other variable is fixed at its values at Φ^*.

in Θ. Structure (40) gives 18 constraints of the type (10), the requirement (4c) gives one constraint. These 19 constraints are reduced to 16 linearly independent constraints, which gives 56-16 = 40 ϕ-parameters. Let all classes be equally probable. In real diagnosis applications the class representing the fault free case is of course often much more probable than the faulty classes, but we use equal probability here to really investigate the effects of the background knowledge.

To exemplify the method, we use two training data sets: the set $\mathcal{D}_1$ with 100 training samples from class *NF* only, and the set $\mathcal{D}_2$ with 50 training samples each from *NF* and F_1. We generate evaluation data sets with 100 samples from the classes *NF*, F_1, and $F_1\&F_2$, and for each data set we compute the average probability assigned to all classes. If, on the average, high probability is assigned to the underlying class, inference is successful. The average probabilities

assigned to the classes, when training data set $\mathcal{D}_1$ is used, is plotted in Figure 3 for evaluation data from *NF* (top), F_1 (middle), and $F_1\&F_2$ (bottom). The result for class F_1 is poor since the corresponding column in (40) is similar to the *NF* column. When data is from $F_1\&F_2$ instead, the difference to data generated from *NF* is larger. Therefore, this class is more easily distinguished. The result when training set $\mathcal{D}_2$ is used is plotted in a similar manner in Figure 4. Finally, for comparison, we have computed the probabilities for the classes when training set $\mathcal{D}_1$ is used without background knowledge. The result is plotted in Figure 5.

Comparing Figures 3 and 4 with Figure 5, we first note that if no background knowledge is used, training data only helps distinguish the class it is generated from, i.e. *NF*. Furthermore, it over estimates the probability for *NF* in the evaluation sets from the other two classes. Using no background knowledge, the classification result is strongly biased by the selection of the experimental training data. By adding background knowledge, more information can be extracted from training data.

In Figure 3 we see that by using background knowledge, inference performance is lost when the underlying class is *NF* compared to when no background knowledge is used. This makes sense, since background knowledge says that all classes are a priori equally likely, and also that elements in the observation vector have the same distribution as under class *NF* under other classes. However, the performance lost when evaluation data is from the same class as training data is regained under other classes, in particular for the case when F_1 is the underlying class.

In Figure 4 it is shown that using training data from both *NF* and F_1 leads to that the true underlying class is among the most probable ones for all three evaluation sets, also for the case $F_1\&F_2$, from which no training data is available. By knowledge about the structure (40), training data from *NF* and F_1 can in fact be reused in learning about all classes except $F_2\&F_3$. This illustrates the fact that the proposed method is able to use background knowledge to improve inference also for classes from which there is no training data.

In the three experiments with results plotted in Figure 3 - Figure 5 we have used the same number of training data, but we have improved diagnosis results in two steps. First, by adding the likelihood constraints given by background knowledge, and second, by collecting data from two classes instead of only one. Since the method handles experimental data, training data collection can be performed in a way so that as much information as possible can be gained from a limited number of samples.

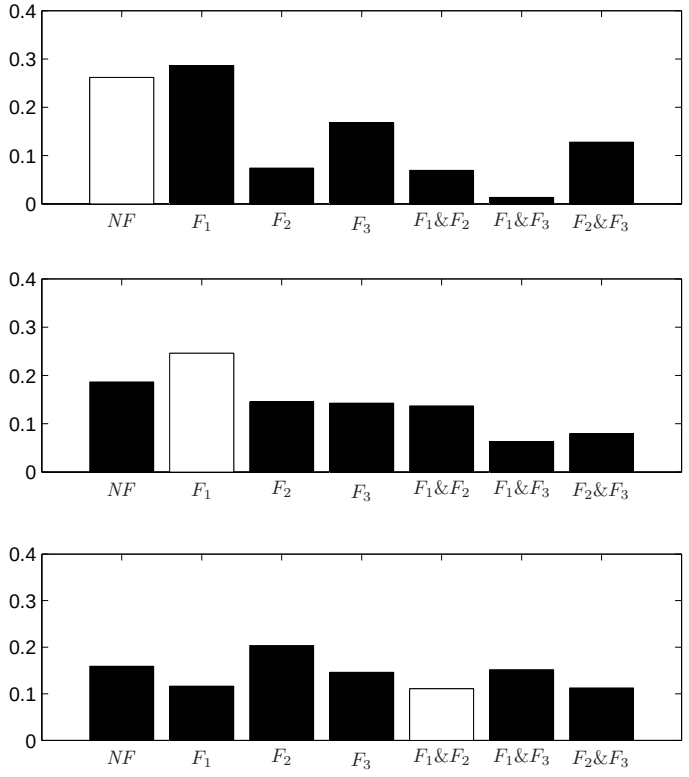


Figure 3: Average probability assigned to different classes when training set $\mathcal{D}_1$ is used. The true underlying class is marked with a white bar.

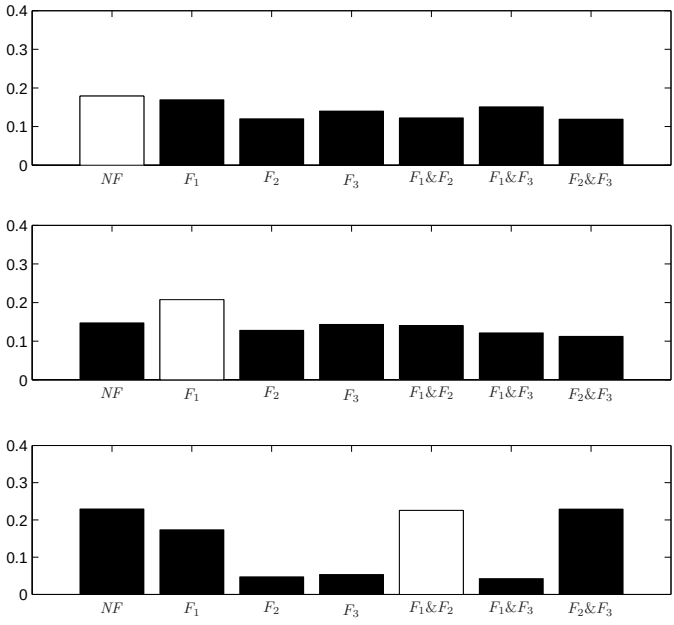


Figure 4: Average probability assigned to different classes when training set $\mathcal{D}_2$ is used. The true underlying class is marked with a white bar.

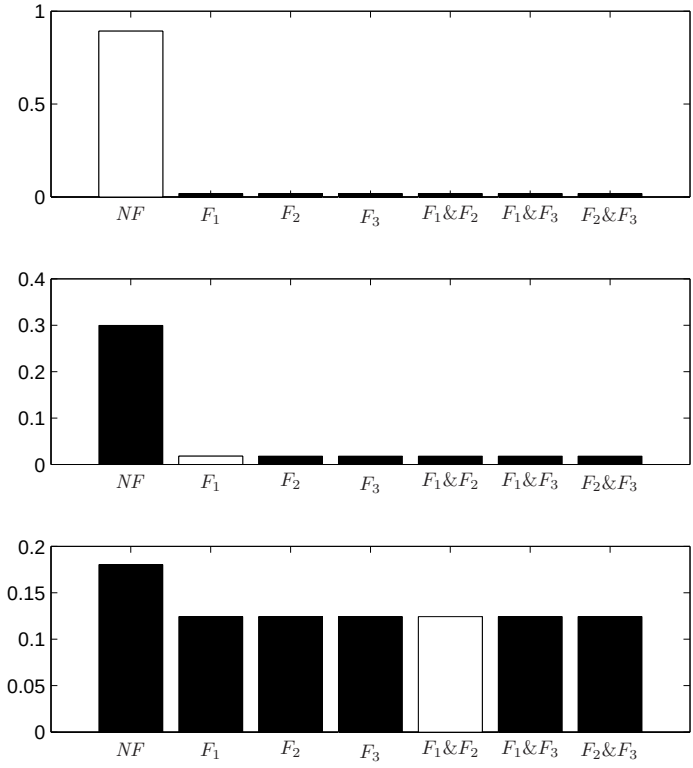


Figure 5: Average probability assigned to different classes when training set $\mathcal{D}_1$ is used without background knowledge. The true underlying class is marked with a white bar.

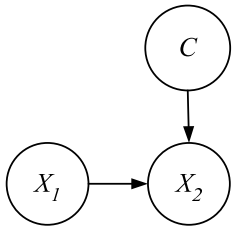


Figure 6: A Bayesian network that satisfies the constraint $p(x_{1i}|c_j) = p(x_{1i}|c_l)$ for all i, j, l.

7 Related Work

We consider the problem of doing inference about one class variable C, given observations $\mathbf{X} = (X_1, \ldots X_R)$ and constraints of the types (11) and (9). The constraints of type (11) can be interpreted as different kinds of independence relations between the class variable and elements in the observation vector. If the equality (11) holds for all values $x_{ik}, k = 1, \ldots K$ of X_i and for all values c_l, $l = 1, \ldots, L$ of C then X_i and C are independent. In this case it would be possible to use a Bayesian network (BN) to describe the dependence relations, and apply traditional learning methods to find the parameters [Heckerman et al., 1995]. For example, the BN is shown in Figure 6 for a case with two observations (X_1, X_2) and one class variable C, and where it is known that X_1 and C are independent. Note, that nothing is known about the dependency relations between X_1 and X_2, and therefore there must be an edge between these two nodes.

If instead the equality (11) holds for all values $x_{ik}, k = 1, \ldots K$ but only for a subset of the values of C, the type of dependencies we study are similar to the context specific independence described in [Boutilier et al., 1996] and can not be represented straightforward by an ordinary BN. In [Boutilier et al., 1996] methods are proposed to extend the concept of BN:s to describe context specific independence.

The two types of independence relations discussed above are special cases of the constraints handled with the method presented in this paper. In addition it also handles the case where equality (11) holds for a subset of the values x_{ik} and a subset of the values of C, which is, to the authors knowledge, not previously considered in literature.

Constrained dependencies similar to the ones studied in the current paper can also be represented in Probabilistic Decision Graphs (PDG) [Jaeger et al., 2005, Jaeger, 2004]. In PDG:s the dependence relations between variables must be tree structured. In [Jaeger et al., 2005] learning PDG:s from data is considered. In the current work we learn parameters and probability distributions

from data and background knowledge. No assumption on independence relations on other variables other than that they should fulfill (16).

In [Giffin and Caticha, 2007] Maximum Entropy methods are proposed for inference problems with background information similar to ours. However, since they rely on Maximum Entropy they consider constraints on expected values rather than on the likelihood as in the current work. In the kind of problems considered in the current paper, application of the Maximum Entropy methods tends to give complex computations and integrals that are difficult to solve. By constraining likelihoods we have in the current paper provided straightforward computations and efficient approximations when needed.

8 Conclusions

We have derived a new method for Bayesian inference from data and background knowledge. The type of background knowledge we study is more general than the background knowledge considered in previous works. It appears in many practical applications, such as fault diagnosis of technical processes, econometrics, and medical diagnosis. Furthermore, the inference method handles experimental training data, which is collected by actively choosing classes from which data is collected.

The background knowledge can be efficiently represented in two parts: a vector of prior probabilities for the classes, and a table, in diagnosis called an FSM. It has been shown how the background knowledge can be translated to likelihood constraints, and then expressed as constraint on the parameters in the computations.

The method derived here results in multidimensional integrals to which there in general are no closed-form solutions. Instead, we have made a detailed description of how to approximate these integrals. Given constraints expressed by a vector and an FSM, the approximation method is easy to implement and inference can be made automatically. The computations have been demonstrated in two examples: a small example investigating the accuracy of the approximation, and an illustrative fault diagnosis example to illustrate the use of the method.

The examples, see e.g. Figure 3, indicate that by combining data and background knowledge, significant improvements in the inference can be made, in particular when the available amount of training data is limited. Furthermore, by comparing Figures 3 and 4, we have seen that we can utilize the experimental data, and collect training data in a way so that as much information as possible can be gained from a limited number of samples.

One challenge in Bayesian inference is the exponential growth of number of parameters as problems become larger. In the current work, the background knowledge reduces the dimensions of matrices and integrals. However, for large

problems with few constraints, there may still be unfeasibly many parameters, and our future work consists in investigating effects of scaling. Future work also includes application of the derived inference method to the diagnosis of a heavy truck engine, using real data.

References

[Boutilier et al., 1996] Boutilier, C., Friedman, N., Goldszmidt, M., and Koller, D. (1996). Context-Specific Independence in Bayesian Networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artifial Intelligence*.

[Daigle et al., 2006] Daigle, M., Koutsoukos, X., and Biswas, G. (2006). Multiple Fault Diagnosis in Complex Physical Systems. In *Proceedings of 17th International Workshop on Principles of Diagnosis (DX 06)*.

[de Kleer and Williams, 1992] de Kleer, J. and Williams, B. C. (1992). Diagnosis with Behavioral Modes. In *Readings in Model-based Diagnosis*, pages 124–130, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Devroye et al., 1996] Devroye, L., Györfi, L., and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer, New York.

[Dharmadhikari and Joag-Dev, 2006] Dharmadhikari, S. W. and Joag-Dev, K. (2006). Multivariate Unimodality. *Encyclopedia of Statistical Sciences*.

[Feelders and van der Gaag, 2005] Feelders, A. and van der Gaag, L. C. (2005). Learning bayesian Network Parameters Under Order Constraints. *International Journal of Approximate Reasoning*, pages 37–53.

[Geiger and Heckerman, 1997] Geiger, D. and Heckerman, D. (1997). A Characterization of the Dirichlet Distribution Through Global and Local Independence. *The Annals of Statistics*, 25(3):1344–1360.

[Giffin, 2007] Giffin, A. (2007). Updating Probabilities: An Econometric Example. In *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag.

[Giffin and Caticha, 2007] Giffin, A. and Caticha, A. (2007). Updating Probabilities with Data and Moments. In *Proceedings of MaxEnt 2007, 27th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*.

[Goutis and Casella, 1999] Goutis, C. and Casella, G. (1999). Explaining the Saddlepoint Approximation. *The American Statistician*, 53(3).

[Heckerman et al., 1995] Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20(3):197–243.

[Jaeger, 2004] Jaeger, M. (2004). Probabilistic decision graphs - combining verification and ai techniques for probabilistic inference. *International Journal of Uncertainty, Fuzziness & Knowledge-Based Systems*, 12:19–42.

[Jaeger et al., 2005] Jaeger, M., Nielsen, J. D., and Silander, T. (2005). Learning probabilistic decision graphs. *International Journal of Approximate Reasoning*, 42:84–100.

[Kontkanen et al., 2001] Kontkanen, P., Myllymäki, P., Silander, T., Tirri, H., and Grünwald, P. (2001). Comparing predictive inference methods for discrete domains. In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, pages 233–238.

[Korbicz et al., 2004] Korbicz, J., Koscielny, J. M., Kowalczuk, Z., and Cholewa, W. (2004). *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Springer, Berlin, Germany.

[MacKay, 2005] MacKay, D. J. C. (2005). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.

[Niculescu et al., 2006] Niculescu, R. S., Mitchell, T., and Rao, R. B. (2006). Bayesian Network Learning with Parameter Constraints. *Journal of Machine Learning Research*, pages 1357–1383.

[Nyberg, 2002] Nyberg, M. (2002). Model-based diagnosis of an automotive engine using several types of fault models. *IEEE Transaction on Control Systems Technology*, 10(5):679–689.

[Pernestål and Nyberg, 2007] Pernestål, A. and Nyberg, M. (2007). Using Prior Information in Bayesian Inference - with Application to Diagnosis. In *Proceedings of 27th international workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (MaxEnt 2007)*.

[Pernestål and Nyberg, 2008] Pernestål, A. and Nyberg, M. (2008). Bayesian Fault Isolation by Combining Data and Process Knowledge with Application to Engine Diagnosis. submitted to *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*.

[Pernestål et al., 2006] Pernestål, A., Nyberg, M., and Wahlberg, B. (2006). A Bayesian Approach to Fault Isolation with Application to Diesel Engine Diagnosis. In *Proceedings of 17th International Workshop on Principles of Diagnosis (DX 06)*, pages 211–218.

[Pulido et al., 2005] Pulido, B., Puig, V., Escobet, T., and Quevedo, J. (2005). A New Fault Localization Algorithm that Improves the Integration Between Fault Detection and Localization in Dynamic Systems. In *Proceedings of 16th International Workshop on Principles of Diagnosis (DX 05)*.

[Reiter, 1992] Reiter, R. (1992). A Theory of Diagnosis From First Principles. In *Readings in Model-based Diagnosis*, pages 29–48, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Sivia, 1996] Sivia, D. S. (1996). *Data Analysis: A Bayesian Tutorial*. Oxford University Press.

Paper 3

Non-stationary Dynamic Bayesian Networks in Modeling of Troubleshooting Processes[1]

Anna Pernestål and Mattias Nyberg

Division of Vehicular Systems, Department of Electrical Engineering,
Linköping University,
Sweden.

Abstract

In research and industry, decision theoretic troubleshooting of complex automotive systems has recently gained increased interest. With suitable troubleshooting, uptime can be increased and repair times shortened. To perform decision theoretic troubleshooting, probability computations are needed. In this work we consider computation of these probabilities under external interventions, which changes dependency relations. We apply a non-stationary dynamic Bayesian network (nsDBN), where the interventions so called events. The events change dependency relations, and drive the nsDBN forward. In the paper, we present how to build models using event driven nsDBN, how to perform inference, and how to use the method in troubleshooting. Event driven nsDBN can be used to model any process subject to interventions, and in particular it opens for solving more general troubleshooting problems than previously presented in literature.

[1]This paper has been submitted to International Journal of Approximate Reasoning. It is partly based on [Pernestål et al., 2009].

1 Introduction

To fulfill performance, safety, and environmental requirements, technical systems become increasingly complex and thus more difficult to diagnose and repair. At the same time, there are requirements on improved diagnosis, shortened repair times, and increased uptime. One of the main approaches to tackle the challenge of shortening repair times of increasingly complex systems is to apply decision theoretic troubleshooting, see for example [Breese and Heckerman, 1996, Langseth and Jensen, 2002, Olive et al., 2003, Warnquist and Nyberg, 2008].

We consider a decision-theoretic troubleshooting-system to consist of two parts: a Planner and a Diagnoser. The Planner searches for the optimal sequence of actions to be applied to make the process fault free, relying on probabilities of faults computed by the Diagnoser. For example, the Planner can use AO^*-search techniques together with heuristics to bound the search space as in for example [Warnquist and Nyberg, 2008, Heckerman et al., 1995]. The reliability of the troubleshooting strategies determined by the Planner is dependent on the accuracy of the probabilities computed by the Diagnoser. In the current work, we focus on the Diagnoser, and its task to compute probabilities of faults and of new observations conditioned on everything known so far.

During troubleshooting, the system is subject to troubleshooting actions. Some of these actions, such as repairs, may change dependencies and conditional probabilities between variables. In this sense, the system is subject to external interventions. In this work we model the system and the troubleshooting process by using a non-stationary dynamic Bayesian network (nsDBN), and use this model to compute relevant probability distributions. In the nsDBN, actions cause events, which in turn generate new time slices. In this sense, the events drive the DBN forward.

The term "non-stationary" is used to highlight the fact that in our dynamic Bayesian network (DBN), the structure of dependencies may differ between different time slices, depending on the actions performed. Using different dependencies in different time slices is in accordance with the general definition of DBNs given for example in [Murphy, 2002] and [Neapolitan, 2003], but is previously only rarely studied in literature [Robinson and Hartemink, 2008].

The main contribution in the paper is the framework for modeling troubleshooting processes by using nsDBNs. This framework makes it possible to deal with more complex system structures and troubleshooting scenarios than in previous works on troubleshooting [Breese and Heckerman, 1996, Langseth and Jensen, 2002, Olive et al., 2003, Warnquist and Nyberg, 2008]. We use nsDBNs to model the troubleshooting process of a system subject to external interventions, and show how to define, represent, and perform inference in these nsDBNs. Both representation and inference is done as compact and storage efficient as possible.

In previous works on decision theoretic troubleshooting, limitations and assumptions on the systems to be troubleshooted are applied to simplify the probability computations. Among the assumptions are single faults [Langseth and Jensen, 2002, Skaanning et al., 2000], the existence of a "problem defining node" which can be directly observed to verify whether the system is fault free [Breese and Heckerman, 1996], or a simple (two-layer) structure of dependencies between observations and components [Warnquist and Nyberg, 2008]. However, when working with real technical systems these assumptions generally do not hold. Systems may have multiple faults when troubleshooting begins, and it is often impossible, or at least very expensive, to verify that the system is fault free. Furthermore, the structure of dependencies is often complex. In the proposed method such assumptions are avoided by applying nsDBNs.

We begin with discussing relations to previous works in Section 2, before presenting the troubleshooting scenario and an example system in Section 3. In Sections 4 - 6 we define, build and perform inference in the event-driven nsDBN. Finally, the theory of nsDBN is illustrated in detail on a troubleshooting example in Section 7, before we conclude in Section 8.

2 Related Work

The use of non-stationary DBN (nsDBN) in modeling for troubleshooting with interventions is related to two research areas: DBNs in general, and interventions in probability theory and Bayesian networks (BN). In this section we give a brief discussion of related work in these two areas.

Previous literature on nsDBN is only sparse. The definition of DBN that we consider, which includes nsDBN, is commonly used in literature, see for example [Murphy, 2002, Neapolitan, 2003]. However, these authors consider only stationary DBN in examples and algorithms. In [Robinson and Hartemink, 2008] the concept of nsDBNs is introduced for modeling dynamic processes with autonomously changing structures, and it is discussed how such models can be learned from data. In the current paper we also consider nsDBN, but the structure changes we consider are effects of external interventions, rather than autonomous changes of the modeled process. Focus in the current paper is on representing and handling these external interventions and their effects.

As an alternative to nsDBNs, non-stationary Markov chains can be used to model non-stationary processes, see for example [Elliott et al., 2001] and [Mamon, 2002] for two financial applications. The main drawback with Markow chains, in contrast to DBN, is that they do not utilize the known structure of probabilistic dependencies. This often leads to unnecessary (and unfeasible) computational burden.

The effects of interventions in BNs in a non-dynamic setting is addressed for example in [Pearl, 2000], [Spirtes et al., 2001], and [Lauritzen, 1999], where

causal Bayesian networks are applied. More recently, interventions in DBNs are studied in [Queen and Albers, 2009]. In all these previous works, interventions are used to identify causal relations, and the close relation between interventions and causality is investigated. There are two main differences between these previous works and the current. The first is the time aspect of the causal relations. In the previous works, causal relations are instantaneous, meaning that the effect of an intervention spreads directly through the network. In the present work on the other hand, dependencies may by caused by causal relations that has been present previously, but are not present at the time for reasoning. For example, in a car, a broken gasket may cause oil to leak out during driving. When the car is parked, there is a dependency between the status of the gasket and the observation that oil has leaked out. If we observe that there is an oil leakage, we can draw conclusions about the gasket. However, if we replace the gasket, there is no longer any dependency between the oil and the gasket until the car has been operated again. The second difference is that in the previous works, there can be external actions intervening with the system. These actions are called *interventions*, and are modeled as active assignment of values to variables in the BN, or as changes of the distribution of variables as in [Queen and Albers, 2009]. In the current work on the other hand, the external actions can be, as previous, assignments to variables or changes in distributions, but they can also be changes in the structure of dependencies between variables.

3 The Troubleshooting Scenario

We consider the following scenario: an automotive vehicle, for example a heavy truck, has entered the workshop. At the workshop, a troubleshooting process begins. During the troubleshooting process symptoms are observed, components are repaired, and the system is possibly run to verify the result of repairs.

In this section we introduce a small sample system that will be used to illustrate the concepts in the remainder of the paper. We show how it can be modeled, and describe the actions that can be applied to it during troubleshooting.

3.1 The OPG System

Consider a small system consisting of a pipe connected to an oil tank via a gasket and a smaller pipe, see Figure 1. The oil is pumped through the system by a pump. We consider the subpart of the system consisting of the pipe, the oil that flows through it, and the gasket. The other parts are assumed to always be fault free. During operation, oil is pumped through the pipe and the gasket. At rest, the system may still be pressurized, but the pump is turned off, so the pressure will not build up. If the oil is of erroneous type, it may be discolored

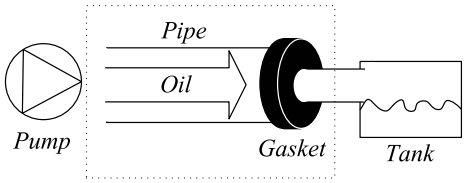


Figure 1: The sample system: oil flows through a pipe that is connected to a smaller pipe via a gasket.

and/or have wrong viscosity. Wrong viscosity may increase the pressure in the pipe. Increased pressure in the pipe may cause a leakage, or cause the gasket to fail which in turn may cause a leakage. A leakage can also be caused by a hole in the pipe. The system in Figure 1 is a typical subsystem of an automotive vehicle. We refer to it as the oil-pipe-gasket (OPG) system, and a model of it is depicted in Figure 2. In the figure, the nodes represent variables in the model of the system[2], and edges represent causal dependencies between the variables. The model comprise three variables representing components: *Oil*, denoted O, *Gasket*, denoted G, and *Pipe*, denoted Pi. Components are marked with gray circles and have two possible values: "faulty" (F) or "non-faulty" (NF). Faulty oil means that it is worn or of erroneous type, a faulty gasket is broken or out of place, and faulty pipe means that there is a hole in the pipe. In the example system there are two observable symptoms. The observable symptoms are marked with squares. The observable symptom *ObservedOilColor*, denoted OOC, can be either *Normal* or *Green*. The observable symptom *ObservedLeakage*, denoted OL, has the possible values *Present* or *NotPresent*. The three remaining variables *OilColor*, (OC), *Pressure*, (Pr), and *Leakage*, (L), are marked with white circles, and are hidden variables that represent internal states in the system. In this example they are all binary and their domains are $OC \in \{Normal, Green\}$, $Pr \in \{Normal, High\}$, and $L \in \{Present, NotPresent\}$.

3.2 Variables

When modeling for troubleshooting we use three types of variables: components, observable symptoms, and internal state variables.

Components. When using the term component we mean both the physical component and a variable, C_i, representing the fault state of the component. A component is a part of the system that can be repaired, i.e. to each component there is a repair action associated, see Section 3.3. The variables C_i are discrete. They always have the possible value "non-faulty", NF. In addition, they have one or several fault states, F_i.

[2]We will use "nodes" and "variables" interchangeably.

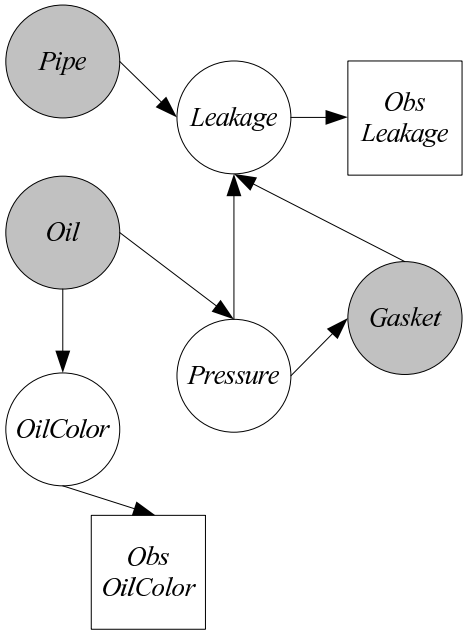


Figure 2: A model of the OPG system consisting of three components (gray circles), two observable symptoms (squares), and three internal states (white circles).

There are two probability distributions associated to each component variable. The first is the prior distribution for the state of the component given the operation history H of the system[3],

$$p(C_i|H). \tag{1}$$

The operation history H represents knowledge about the operation history and usage of the system. For example, in systems that has been operated a long time on an exceptionally high load, the prior probability for faults will typically be higher than for a system operated at normal load. In modern automotive vehicles, knowledge about the operation history is stored in the on-board control units. The distribution (1) can for example be learned from fleet operation data, expert knowledge and experience, or component specifications. However, in the current work, we ignore the operation history to keep notation free from clutter, and assume that $p(c_i|H) = p(c_i)$.

The second distribution related to components is

$$p(C_i|repair(C_i)), \tag{2}$$

describing the probability that a repair of component C_i actually resulted in fault free component. In previous works on decision theoretic troubleshooting it is often assumed that repairs are always successful [Heckerman et al., 1995, Langseth and Jensen, 2002, Warnquist et al., 2009], and repairs are thus modeled by adding evidence to the component variables. In the current work we assume that components are not directly observable, meaning that evidence can not be added to them. Repairs that are always successful are modeled by using the distribution $p(C_i = NF|repair(C_i)) = 1$ for (2). Of course, in the computations, this is equivalent to adding evidence to the component variable.

Observable Symptoms. Observable symptoms, or observation variables, O_i, are discrete or continuous variables representing observations that can be made of the troubleshooted system. These variables state the only ways through which the system can be monitored. In the current work we restrict us to use only discrete observable symptoms. The same theory as described in the paper applies to the continuous case, but inference techniques become more complicated

Internal Variables. To model a system with sufficient precision, it may be necessary to use variables representing internal states of the system. The internal variables may be discrete or continuous, and can not be directly observed. In Figure 2 we have distinguished between the internal state OC and the observable symptom OOC. This construction highlights the fact that even if the system has a specific internal state, it is not necessarily the case that the true state is observed. For example, although the oil color is green, it may

[3]We use $p(X)$ to denote the distribution of the variable X, and $p(X = x)$ or $p(x)$ to denote the probability that X takes the value x.

be the case that the mechanic observes normal oil color. As for the observables symptoms, we restrict the current work to only consider discrete internal variables to keep inference less complicated.

3.3 Troubleshooting Actions

During troubleshooting, there are external interventions with the systems in terms of *troubleshooting actions* are applied to the system. These actions are modeled by changing dependencies between variables, by changing parameters in probability distributions, or by adding evidence to variables. We consider three types of troubleshooting actions: repairs of components, observations of observable symptoms, or operation of the system. In this section we describe the characteristics of the actions, while they will be formally defined in Section 4.

Repair Actions. A repair action is applied to a component variable, and the repair of component C_i is denoted $repair(C_i)$. The probability that the repair is successful is determined by the distribution (2). A repair action typically remove dependencies related to the repaired component. It also updates the probability distribution for the repaired component.

Observation Actions. An observation action is applied to an observable symptom variable, $observe(O_k)$. This action simply means adding evidence to the observable symptom variable.

Operation Actions. The action to operate the system for a certain time τ is denoted $operate(\tau)$. The operation action affects the complete systems by introducing dependencies between variables and changing parameters in the probability distributions.

3.4 Actions, Evidence, and Events

The troubleshooting actions describe the troubleshooting process. An action results in *evidence* and/or *events*. Evidence is an assignment of known values to a subset of the variables, and it is the observation actions that result in evidence. An event is a change in the structure of dependencies between the variables. The repair and operation actions results in events.

4 Dynamic Bayesian Networks

We will now provide the definitions of BN and DBN used in the paper.

4.1 Definitions of BN and DBN

We use the definition of Bayesian networks given in [Jensen and Nielsen, 2007].

Definition 1 (Bayesian Network)**.** *A* Bayesian network *(BN) is a triple* $B = (\mathbf{X}, \mathcal{E}, \Theta)$*, where* $\mathbf{X}$ *is a set of variables with a finite set of mutually exclusive states, and* $\mathcal{E}$ *is a set of directed edges between the nodes. The nodes and the directed edges form a directed acyclic graph* Γ*. The set* Θ *are parameters defining the conditional probabilities* $P(X_i|pa(X_i))$*, where* $pa(X_i)$ *are the parents of* X_i *in* G*.*

The BN B represents the joint probability distribution

$$P(\mathbf{X}) = \prod_{X_i \in \mathbf{X}} P(X_i|pa(X_i)).$$

A remark on notation: we use the convention that if a BN is denoted with a sub- and/or superscript, the sets of variables, edges, and parameters are equipped with the same sub- and/or superscript. For example, B^0 has variables $\mathbf{X}^0$, edges $\mathcal{E}^0$, and parameters Θ^0.

When considering processes that change over time, as in modeling for troubleshooting, one approach is to use a DBN. There are a few slightly different, but similar, definitions of DBN, see e.g. [Neapolitan, 2003, Russell and Norvig, 2003, Jensen and Nielsen, 2007]. In the current work we take a general view of DBN, and define it as follows.

Definition 2 (Dynamic Bayesian Network)**.** *A* dynamic Bayesian network *(DBN) is a BN where the nodes can be partitioned in sets of nodes* $\mathbf{X}^0, \mathbf{X}^1, \ldots$ *such that for an* l*:th order DBN, each node* $X_i^t \in \mathbf{X}^t$ *only have parents in the sets* $\mathbf{X}^{t-m}, m = 0, \ldots, l$*.*

In particular, we will only use 1:st order DBN, i.e. with $l = 1$. This simplifies notation. Extension of expressions and results to $l > 1$ is straight-forward. In the DBN, the nodes $\mathbf{X}^t$ and the edges in between them form a Bayesian network B^t. Using the nomenclature from [Jensen and Nielsen, 2007], the BN B^t is called a "time slice t". Time slices are connected via *temporal links*. An example of a DBN is shown in Figure 3.

In our definition of DBN, it is possible to have different number of nodes in each time slice. However, in the current paper we restrict us to only consider DBN with the same number of variables in each time slice, and let each variable X_i^t represent the evolution of some quantity in the system that is modeled. On the other hand, note that the structure of dependencies within and between two time slices may change. We say that the DBN is *non-stationary*. They most commonly used definitions of DBN in literature, including several standard works as for example [Murphy, 2002, Russell and Norvig, 2003, Neapolitan, 2003], allows non-stationary DBNs. However, in examples, applications, and algorithms in theses references only stationary DBNs are considered. In a stationary DBN, time slices and temporal links are equal for all t.

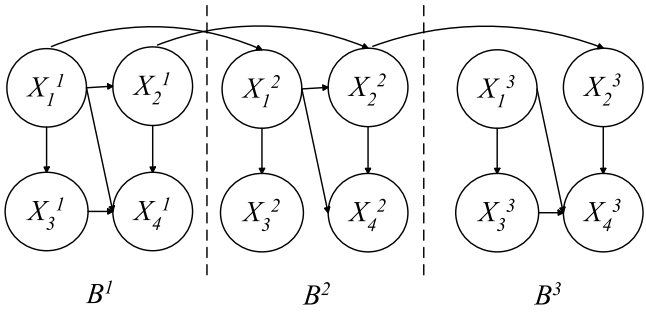


Figure 3: A DBN with three subgraphs (time slices).

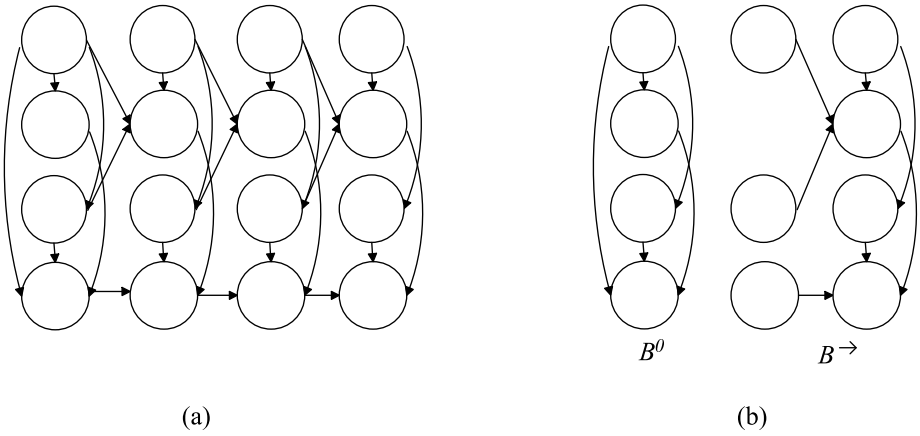


Figure 4: (a) A stationary DBN (b) The initial BN B^0 and the transition BN $B^{\rightarrow}$ needed to characterize the DBN in (a).

4.2 Characterizing an nsDBN

As described in for example [Murphy, 2002] and [Neapolitan, 2003] a DBN modeling a stationary process (i.e. a stationary DBN) is fully described by an *initial* BN B^0 and a *transition* BN $B^{\rightarrow}$, as shown in Figure 4. The initial BN describes the system at the initial time, while the transition BN describes the relations between two time slices and within one time slice. In an nsDBN, on the other hand, the transition BNs can be different at different times, see Figure 3. We use $B^{\rightarrow t}$ to denote the transition BN from time $t-1$ to t.

5 Building Non-stationary DBN Driven by Events

In this section, we make the characterization of nsDBN, and in particular the initial BN and the nominal transition BN, more concrete. As described in Section 3.3 is the troubleshooting process described by troubleshooting actions. In Section 3.4 it was described how these actions may result in events which change the dependency relations between parts of the system. We model the troubleshooting process by using an nsDBN, where the events generates new time slices. We say that the nsDBNs we consider are driven by the events. In this way, each time slice models the system between two events. Using the nomenclature by [Robinson and Hartemink, 2008], we call the time interval between two events an *epoch*. We assume that the system is stationary in an epoch, and thus each epoch is represented by one time slice. During an epoch variables may be observed.

To make the characterization from the previous section more precise we use the following three information components:

1) An initial BN, B^0, modeling the system at the initial time.

2) A *nominal transition BN*, $B^{\rightarrow t}_{nom}$, representing the transition BN under the empty event, i.e. when there are no external interventions.

3) For each event, e, knowledge about how it differs compared to the nominal transition BN.

5.1 Initial BN

As for the stationary case, the initial BN, B^0, models the system at the initial time. The initial BN can for example be learned from training data and/or by expert knowledge about the system.

5.2 Nominal Transition BN

The nominal transition BN, $B^{\rightarrow t}_{nom}$, is the transition BN from $t-1$ to t obtained by the empty event, when there are no interventions with the system. The nominal transition BN is hypothetical, since in practice there is no need to generate a new time slice when the event is empty. It is used as reference, and the effects of all other events are defined as differences compared to the nominal transition BN.

As superscript t indicates, $B^{\rightarrow t}_{nom}$ may be different depending on the structure of the previous time slice. Thus, it must be defined in relation to the previous time slice. To construct the nominal transition BN we use the following classification of edges and variables (nodes).

Definition 3 (Instant Edge)**.** *An edge within a time slice (i.e. in between nodes in* $\mathbf{X}^t$*) is* instant *if it represent a causality relation that is still present after the empty event, i.e. the event where there are no external interventions. The change of value of the parent node has instantaneous influence on the child node. An edge that is not instant is* non-instant*.*

For example, in the OPG-system, erroneous O will affect the internal variable OC instantly, and the edge between the nodes is instant. A non-instant edge represents a dependency relation that needs operation of the system to be visible. An example of a non-instant edge in the OPG-system is the one between O and Pr, since the system need to be operated for the oil to cause the pressure to rise.

Definition 4 (Persistent Variable)**.** *A variable is* persistent *if its value at time* t *is dependent on its value at time* $t-1$ *under the empty event. A variable that is not persistent is* non-persistent*.*

For example, component variables are typically persistent, since they are known to be the same if no action is applied. Observed symptoms, on the other hand, are typically non-persistent. For example, although a leakage is observed at time $t-1$, there is no guarantee that it will be observed at time t as well.

When constructing a (nominal) transition BN is the *outgoing interface* an important set of variables. It consists of the set of variables in $\mathbf{X}^{t-1}$ that have children in $\mathbf{X}^t$.

$$\mathbf{I}^{\rightarrow t} = \{X^{t-1} \in \mathbf{X}^{t-1} : \exists X^t \in \mathbf{X}^t, (X^{t-1}, X^t) \in \mathcal{E}^{\rightarrow t}\}, \tag{3}$$

The nominal transition BN $B^0_{\rightarrow t}$ can now be constructed by performing the following steps.

(a) Copy the variables $\mathbf{X}^{t-1}$ (not their values) from time slice $t-1$ to time slice t. Label the new variables $\mathbf{X}^t$.

(b) For all instant edges between nodes in $\mathbf{X}^{t-1}$, add a corresponding edge in $\mathbf{X}^t$.

(c) For all persistent variables X_i^{t-1}, add a temporal edge between X_i^{t-1} and X_i^t.

(d) The nodes $\mathbf{I}_t^{\rightarrow} \cup \mathbf{X}^t$ now constitutes the nominal transition BN $B^{\rightarrow t}_{nom}$.

5.3 Effects of Events

Repair and operation actions result in events, and when an event occurs a new epoch begins and a time slice should be added to the nsDBN, i.e. a transition BN should be constructed. Let $B_e^{\rightarrow}$ be the transition BN caused by event e.

We define an event e by the differences between $B_e^{\rightarrow}$ and the nominal transition BN $B_{nom}^{\rightarrow t}$. Thus, an event is defined by the sets of edges added to and removed from $B_{nom}^{\rightarrow t}$. To complete the definition we also need the CPDs that are different from $\Theta_{nom}^{\rightarrow t}$. An event is determined by the following three sets.

- The set $\mathbf{A}_e = \{(X, Y) \notin \mathcal{E}_{nom}^{\rightarrow t}\}$ of edges to be added to $B_{nom}^{\rightarrow t}$.
- The set $\mathbf{R}_e = \{(X, Y) \in \mathcal{E}_{nom}^{\rightarrow t}\}$ of edges the be removed from $B_{nom}^{\rightarrow t}$.
- The set Θ_e of parameters (or, similarly, CPDs) that are different from the parameters $\Theta_{nom}^{\rightarrow t}$.

As described in Section 3.3 repair actions remove dependencies, and thus the set $\mathbf{A}_e$ will typically be empty for repairs. Operation actions on the other hand, add dependencies, and for operations the set $\mathbf{R}_e$ will be empty. This will be further exemplified in Section 7. The set Θ_e includes CPTs for the variables to which incoming arcs are removed, or to which incoming arcs are added. Furthermore, Θ_e may also include CPTs for variables where the dependency structure is not changed, but where only the strength of the dependency is changed by the event. As illustrated in Section 7, this is typically the situation for operation actions.

Remark 1 (Modeling Dynamics)**.** Above, we have assumed that the system is static during each epoch, and that it is only events that may change the system. In some situations this assumption is a restriction. For example, in the OPG system, if there is a hole in the pipe, we might want to model how the oil flows through this hole, resulting in a continuously decreasing pressure. In the event-driven nsDBN framework this can be modeled by instead of using a static BN in each epoch, using an ordinary stationary DBN.

6 Inference in Event Driven non-stationary DBN

Given the initial BN, the nominal transition BN, and the sets $\mathbf{A}_e$, $\mathbf{R}_e$, and Θ_e defining the events as described in the previous section, a sequence of troubleshooting actions $\mathbf{a}^{1:t}$ will completely determine an nsDBN with t time slices, or epochs. In this section, we will discuss how inference can be performed in such an nsDBN.

6.1 A Recursive Inference Algorithm

Given the nsDBN characterized as in Section 5, we search probability distributions of the type

$$p(\mathbf{Z}^t|\mathbf{a}^{1:t}), \tag{4}$$

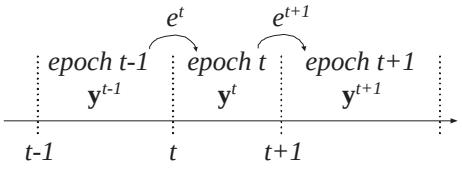


Figure 5: Three epochs with observations in the epochs. Evidence $\mathbf{y}^t$ is added in the epochs, and events start new epochs.

where $\mathbf{Z}^t \subseteq \mathbf{X}^t$, and $\mathbf{a}_{1:t}$ are all actions performed up to and including time t. Repair and operating actions are performed at certain times, $1, \ldots, t$, and, as described in Section 3 they give rise to events. An event starts a new epoch, i.e. generates a new time slice. Let e^t be the event that starts epoch t, and thus generates time slice t, see Figure 5. With this convention time slice t describes the system between times t and $t+1$. The collection of all events from time 1 to time t are denoted $\mathbf{e}^{1:t} = (e^1, \ldots e^t)$. Observation actions are performed in the time interval between two events, i.e. in the epochs. An observation action in epoch t, i.e. in the time interval between t and $1+1$, adds evidence to a subset $\mathbf{Y}^t \subseteq \mathbf{X}^t$ of variables. This means that event e^t occurs before evidence $\mathbf{y}^t$ is collected.

Separating the actions in evidence and events we can rewrite (4) as

$$p(\mathbf{Z}^t|\mathbf{y}^{1:t}, \mathbf{e}^{1:t}), \tag{5}$$

where $\mathbf{y}^{1:t} = (\mathbf{y}^1, \ldots, \mathbf{y}^t)$. This distribution can be obtained from $p(\mathbf{X}^t|\mathbf{y}^{1:t}, \mathbf{e}^{1:t})$ by marginalization over the variables $\bar{\mathbf{Z}}^t = \mathbf{X}^t \setminus \mathbf{Z}^t$ that are in $\mathbf{X}^t$ but not in $\mathbf{Z}^t$. Therefore, without loss of generality, we assume that $\mathbf{Z}^t = \mathbf{X}^t$ to simplify notation.

We now separate the most recent evidence and event, and apply Bayes' rule to obtain

$$\begin{aligned} &p(\mathbf{X}^t|\mathbf{y}^{1:t}, \mathbf{e}^{1:t}) = \\ &= \alpha p(\mathbf{y}^t|\mathbf{X}^t, \mathbf{y}^{1:t-1}, \mathbf{e}^{1:t}) p(\mathbf{X}^t|\mathbf{y}^{1:t-1}, \mathbf{e}^{1:t}) = notag \qquad (6) \\ &= \alpha p(\mathbf{y}^t|\mathbf{X}^t, e_t) p(\mathbf{X}^t|\mathbf{y}^{1:t-1}, \mathbf{e}^{1:t}), \qquad (7) \end{aligned}$$

where α is a normalization constant and we, in the second equality in (7), have noted that $\mathbf{y}^t$ is conditionally independent from $\mathbf{y}^{1:t-1}$ and $\mathbf{e}^{1:t-1}$ given $\mathbf{X}^t$. Since $B^{\to t}$ is determined using the methods from Section 5 we have

$$p(\mathbf{y}^t|\mathbf{X}^t, \mathbf{e}^t) = p(\mathbf{y}^t|\mathbf{X}^t, B^{\to t}).$$

To compute the second distribution of (7), marginalize over the variables in the

past time slice:

$$p(\mathbf{X}^t|\mathbf{y}^{1:t-1},\mathbf{e}^{1:t}) = \sum_{\mathbf{x}^{t-1}} p(\mathbf{X}^t|\mathbf{x}^{t-1},\mathbf{y}^{1:t-1},e^t,\mathbf{e}^{1:t-1})p(\mathbf{x}^{t-1}|\mathbf{y}^{1:t-1},e^t,\mathbf{e}^{1:t-1}). \tag{8}$$

In the first probability in this sum use that the variables $\mathbf{x}^{t-1}$ d-separates $\mathbf{X}^t$ from all variables before $t-1$, and thus e^t is enough to define the relations between $\mathbf{X}^{t-1}$ and $\mathbf{X}^t$,

$$p(\mathbf{X}^t|\mathbf{x}^{t-1},\mathbf{y}^{1:t-1},e^t,\mathbf{e}^{1:t-1}) = p(\mathbf{X}^t|\mathbf{x}^{t-1},e^t). \tag{9}$$

In the second probability in the sum in (8), we note that e^t defines the relations to the children in time slice t. These children are barren nodes [Jensen and Nielsen, 2007], i.e. they will never be assigned any values in the distribution we are to compute, and therefore they will never effect it. This gives

$$p(\mathbf{x}^{t-1}|\mathbf{y}^{1:t-1},e^t,\mathbf{e}^{1:t-1}) = p(\mathbf{x}^{t-1}|\mathbf{y}^{1:t-1},\mathbf{e}^{1:t-1}). \tag{10}$$

By using (8), (9), and (10) we can write (7) as

$$p(\mathbf{X}^t|\mathbf{y}^{1:t},\mathbf{e}^{1:t}) = \alpha p(y^t|\mathbf{X}^t,e^t)\sum_{\mathbf{x}^{t-1}} p(\mathbf{X}^t|\mathbf{x}^{t-1},e^t)p(\mathbf{x}^{t-1}|\mathbf{y}^{1:t-1},\mathbf{e}^{1:t-1}).$$

Furthermore,

$$p(y^t|\mathbf{X}^t,e^t) = p(y^t|\mathbf{X}^t,B^{\to t}),$$
$$p(\mathbf{X}^t|\mathbf{x}^{t-1},e^t) = p(\mathbf{X}^t|\mathbf{x}^{t-1},B^{\to t}),$$

and the inference problem reduces to computing $p(\mathbf{X}^t|\mathbf{y}^{1:t},\mathbf{e}_{1:t})$ from the distribution $p(\mathbf{X}^{t-1}|\mathbf{y}^{1:t-1},\mathbf{e}_{1:t-1})$. We also note that it is sufficient to have only one active transition BN at a time.

6.2 Frontier and Interface Algorithms

The algorithm derived above is based on the fact that $\mathbf{X}^t$ d-separates the variables before time slice t from the variables after t. Instead of using $\mathbf{X}^t$ as separator, a set $\mathbf{F}^t$, called the *frontier*, can be used, see [Murphy, 2002]. In [Murphy, 2002] Murphy shows that using the frontier reduces the dimension of the probability computations needed. He also presents two simple rules that guarantee that the frontier evolve forward.

Another approach is to use the outgoing interface $\mathbf{I}^{\to t}$, defined by (3), as the d-separating set, see [Murphy, 2002]. One main advantage with using the outgoing interface as the d-separating set in stationary DBN is that it facilitates efficient construction of a junction tree that can be reused for all time slices. Unfortunately, in the non-stationary case this advantage is lost.

7 Application to Troubleshooting

We demonstrate the use of event driven nsDBN in troubleshooting by applying the method to the OPG system introduced in Section 3. There are two phases: the preparation phase, where the nsDBN is defined using the methods from Section 5; and the inference phase, where the troubleshooting is performed.

7.1 Preparation: Building nsDBN for Troubleshooting

In the preparation phase the model of the system and the troubleshooting process is build. This is, like most modeling tasks, an artwork and requires knowledge about the system. As an alternative (or complement) to system knowledge, training data can be used to learn the models. However, for event driven nsDBN a lot of data is needed since there are several BNs to learn. Furthermore, in the troubleshooting application, the troubleshooting should function when the system is newly released at the market - before any training has been collected.

In this section we illustrate how modeling may be done, and structure the modeling task. The preparation phase consists of three steps:

1) Building the initial BN B^0.

2) Building the nominal transition BN $B^{\rightarrow t}_{nom}$.

3) Defining the events.

1) Initial BN. The troubleshooting process begins with the system's arrival at the workshop. At this stage, the system is described by the initial BN B^0. Most often we will assume that the system has been operating during a sufficiently long time for all (non-instant) dependencies to fully establish. In terms of the operation action this means that before troubleshooting $operate(\tau_0)$, were $\tau_0 > T$ for a sufficiently large constant T, has been applied. The initial BN B^0 for th OPG system is shown in Figure 2. The parameters Θ^0 are assumed to be known. They can for example be given by experts or learned from data.

2) Nominal Transition BN. The nominal transition BN model the system under the empty event, i.e. when the system is at rest at the workshop. To build the nominal transition BN of the OPG system, we begin with two copies of the nodes, arranged in two time slices. Now, edges within the a time slice should be classified as instant or not, and variables as persistent or not. All non-instant edges should be removed in the second time slice.

When the system is at rest at the workshop, there will be no oil flow so the oil can not cause a pressure build-up, meaning that the (O, Pr) edge is non-instant. Also, the (Pr, G) edge is non-instant, since we have assumed that components can not brake during rest at the workshop. All other edges within the second time slice are instant, and remain the same in the second time slice as in the first. The classification of all the edges, including short motivations,

Table 1: Classification of the edges in the OPG system.

Edge	Number	Type	Motivation
(O, OC)	9,13	instant	The oil color changes immediately when the oil is replaced.
(O, Pr)	8	non-instant	The pressure can only be built-up during operation.
(OC, OOC)	16	instant	The oil color is immediately observable.
(Pr, G)	10	non-instant	Operation is needed for components to change status.
(Pr, L)	14	instant	There may be pressure in the pipe also at rest.
(G, L)	6,15	instant	There is oil in the pipe, both at rest and during operation.
(Pi, L)	5,11	instant	There is oil in the pipe, both at rest and during operation.
(L, OL)	12	instant	A leakage is immediately visible.

is given in Table 1. The numbers in the edge column refers to the numbers in Figure 6.

To determine the temporal links between the two time slices, the variables should be classified as persistent or not. For all persistent variables there should be a temporal link between its copies in the two time slices.

Components will not change spontaneously, and thus they are persistent. The pressure may is not necessarily the same as in the previous time slice, but it is dependent on the its previous value. In particular, the pressure can not increase when the system is at rest at the workshop. Thus the variable Pr is persistent. The variable OC is also persistent. If the oil has a certain color, it will have the same color until changed. However, if two mechanics observe the oil at different times, they do not necessarily do the same observation of the color. This is modeled by making the variable OOC non-persistent. The variables L and OL are also non-persistent. In particular, for the variable L it

Table 2: Classification of the variables in the OPG system.

variable	type	motivation
O	persistent	Components do not change spontaneously.
G	persistent	– " –
Pi	persistent	– " –
OC	persistent	The oil do not change color spontaneously.
Pr	persistent	At rest, the pressure is dependent on its previous value.
L	non-persistent	Although there is a hole, it is not necessarily a leakage.
OOC	non-persistent	Observed symptoms are independent.
OL	non-persistent	– " –

is the case that although there may be a hole in the pipe or a broken gasket, there will not necessarily be a leakage through it all the time. A summary of the classification of the variables is presented in 2. The classifications of edges and variables in Tables 1 and 2 give the nominal transition BN shown in Figure 6.

3) Defining Events. The next step in the preparation phase is to define the events e by determining the sets $\mathcal{A}_e$, $\mathcal{B}_e$, and Θ_e. For the OPG system, the sets are listed in Table 3. The numbers in the two middle columns refers to Figure 6. After operation, non-instant dependencies between components are recovered, and edges are inserted. As a rule-of-thumb, the edges and dependencies in the initial BN are recovered after operation. However, this is not always the case, since the strength of the dependencies may depend on the operation time τ. In Table 3 we use a subscript τ on the distributions Θ_e corresponding to the event $operate(\tau)$, to highlight that they are dependent on the operation time.

For the repair actions, Θ_e consists of the CPDs related to the removed edges only. For the operation action, Θ_e consists of both the CPDs related to the added edges, but there are also new CPDs for the components although the edges are the same. The reason is that at rest at the workshop, the components can not suddenly fail (nor suddenly become fault free). For $C \in \{O, G, Pi\}$, we use the following CPT in the nominal transition BN:

$p(C^t \mid C^{t-1})$	$C^t = NF$	$C^t = F$
$C^{t-1} = NF$	1	0
$C^{t-1} = F$	0	1

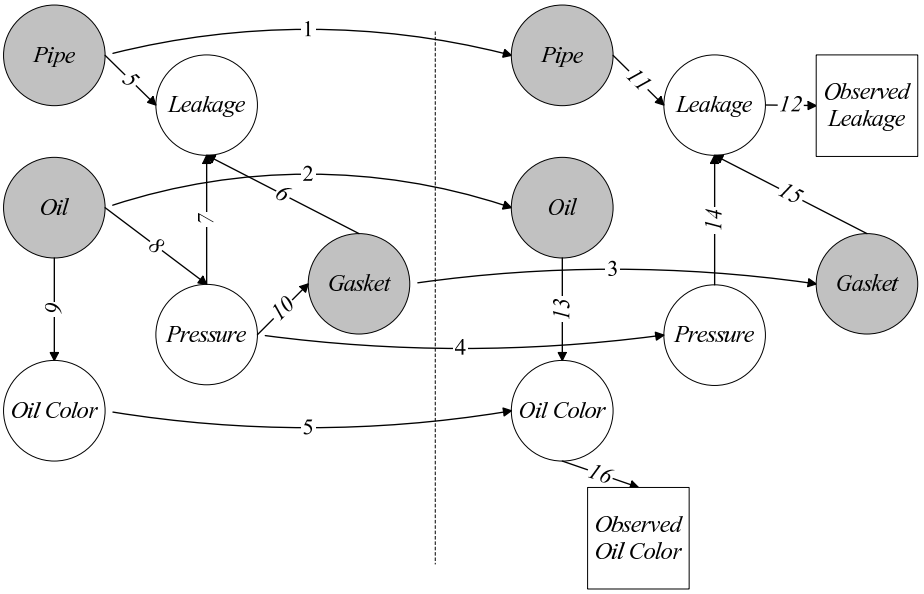


Figure 6: The nominal transition BN for the OPG example.

During operation, the components may fail, and we apply the CPT:

$p(C^t \mid C^{t-1})$	$C^t = NF$	$C^t = F$
$C^{t-1} = NF$	1	0.01
$C^{t-1} = F$	0	0.99

In Table 3 we see that repair actions remove edges, while the operating action add edges. Repeated use of repair events will make the parts of the system less dependent and isolate components from each other. Operation of the system recover non-instant edges (dependencies), and makes it possible to use these dependencies to verify the states of the components. For example, if the gasket is replaced in the OPG system, it is necessary to operate the system after the repair to be able to verify the result of the repair. There are also more intricate situations where non-instant edges may be useful in troubleshooting, see the following example.

Example 7.8 (Indirect Reasoning).
A system has three components, C_1, C_2, and C_3, where the two first component are complex and expensive, while the third is cheap and easily replaced. Assume that faults in C_1 and C_2 gives the same symptoms (and observations), but C_1 may cause C_3 to fail during operation, while C_2 never do. Since $p(C_3 = NF | repair(C_3)) \approx 1$, we can distinguish between faults in C_1 and C_2 by replacing C_3, operate the system, and then investigate the status of C_3.

Table 3: List of events in the OPG system.

Event	$\mathbf{A}_e$	$\mathbf{R}_e$	Θ_e	Motivation
$repair(O)$	-	2,5	$p(O)$	The new oil has new oil color.
$repair(G)$	-	3,4	$p(G)$	The pressure is reset when the gasket is replaced.
$repair(Pi)$	-	1,4	$p(Pi)$	The pressure is reset when the pipe is replaced.
$operate(\tau)$	(O, Pr) (Pr, G)	-	$p_\tau(O^t \mid O^{t-1})$ $p_\tau(Pi^t \mid Pi^{t-1})$ $p_\tau(G^t \mid Pr^t, G^{t-1})$ $p_\tau(Pr^t \mid O^t)$	Non-instant edges are recovered.

7.2 Inference: Computing Probabilities

To illustrate inference in event driven nsDBN, we compute the marginal probabilities for faults in the three components in the OPG system during a troubleshooting sequence. We have implemented the event driven nsDBN for the OPG system in Matlab, using BNT Toolbox by [Murphy, 2001]. The program takes a sequence of actions as input. It returns the probabilities for the three faults at given times. We use the following sequence of observations and repairs:

(o1) $observe(OL = Present)$

(r1) $repair(Pi)$

(o2) $observe(OL = Present)$

(r2) $repair(G)$

(op) $operate(\tau)$

(o3) $observe(OL = NotPresent)$

This sequence gives four time slices in the nsDBN, shown in Figure 7. In the figure are observed nodes marked with dotted frames. The probabilities for faults during the troubleshooting process are given in Table 4.

The sequence describe the following scenario. Consider a truck is driving on the road. The driver experiences troubles with the braking system, and decides to drive to the workshop. At the workshop, a computer aided troubleshooting system, that computes probabilities of faults, is connected to the truck. When the truck arrives to the workshop, the system is described by the initial BN, and is in epoch 0. A leakage is observed. This add evidence to OL^0. Given this first observation, the probabilities for faults are computed. The probability of fault is largest for the pipe, see Table 4. Thus, the workshop mechanic decides to repair the pipe, starting epoch 1 in Figure 7. As listed in Table 3, the edges (Pi^{t-1}, Pi^t) and (Pr^{t-1}, Pr^t) are removed. After the repair, the leakage is observed to be still present, and the gasket is now the most probable component to be faulty. The mechanic replaces the gasket, which begins epoch 2, and then operates the system. After operation the structure within epoch 3 returns to the initial structure. In Table 4, we see that during operation the probabilities for faults increase, but after the final observation, that the leakage is no longer present, the probabilities decrease again.

In Table 4 we see that the probability for O is almost the same during the whole troubleshooting. The reason is that at the workshop, the dependency relations between O and OL are very weak, and only goes through the first epoch. Furthermore, it can be seen how repair actions remove dependencies, while the operation action adds dependencies.

In this small troubleshooting example, we have seen how probabilities a troubleshooting process can be modeled using only the nominal transition BN, the initial BN, and the three sets defining the events. With this information, any sequence of actions can easily be applied and probabilities computed. We have computed probabilities for faults, but probabilities for observations and internal variables can equally easily be computed. The method can be applied to any kind of system, and no assumptions on single faults, no special dependency structure assumptions, nor any function verification node is needed.

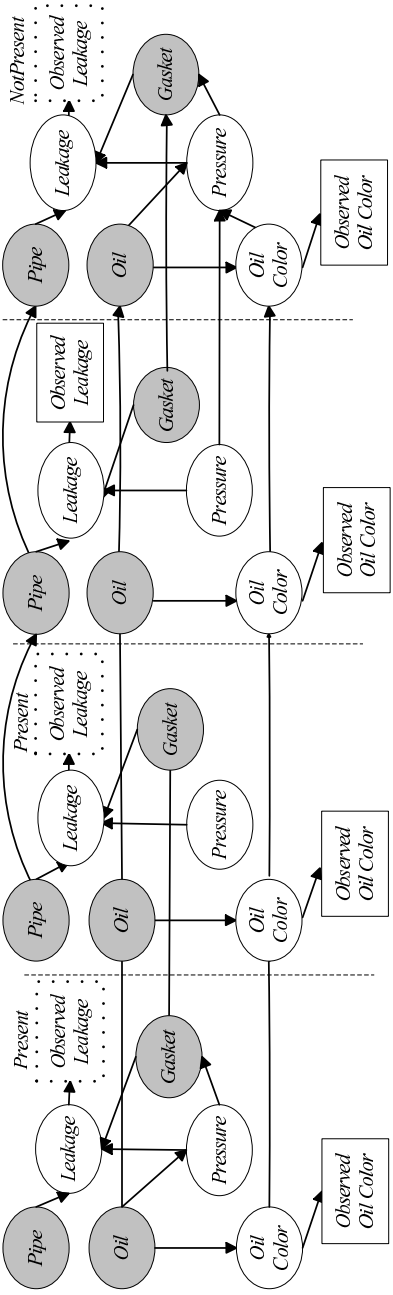


Figure 7: An nsDBN for the three actions $repair(Pi)$, $repair(G)$, and $operate(\tau)$.

Table 4: Probabilities for faults after observations, enumerated as above.

After observation	$p(O = G\|\mathbf{a}_{1:i})$	$p(G = G\|\mathbf{a}_{1:i})$	$p(Pi = G\|\mathbf{a}_{1:i})$
(o1)	0.1523	0.3499	0.5357
(r1)	0.1523	0.3499	0.01
(o2)	0.185	0.8082	0.0238
(r2)	0.185	0.01	0.0238
(op)	0.1858	0.0637	0.0248
(o3)	0.1794	0.0262	0.0102

8 Conclusions

We have considered modeling of troubleshooting processes, and in particular how to handle external interventions that changes the dependencies between components in the system under troubleshooting. To model such processes, we have proposed a new type of Bayesian networks, called event driven non-stationary dynamic Bayesian networks (nsDBN). An nsDBN is a DBN where the dependency structure between variables in a time slice changes over time. The structure changes are caused by events, resulting from external intervention with the system through actions. We have shown how an event driven nsDBN can be fully characterized by an initial BN, a nominal transition BN, and the set of possible events. Each event is compactly defined by three sets.

Throughout the paper we have motivated, and exemplified the event driven nsDBN on the task of modeling for troubleshooting technical systems. In troubleshooting, the system is subject to observations, and repair and operation actions. While observation actions generate evidence, repair and operation actions generate events.

In the paper we have shown how to characterize the nsDBN compactly, and how this characterization can be used to perform inference during any sequence of troubleshooting actions. Building nsDBNs, as modeling in general, is an artwork, but we have provided a structured way of doing this. We have also illustrated the use of nsDBNs on a small troubleshooting example, where the steps of both modeling and inference are shown in detail.

With this paper, we have taken a first step towards using event driven nsDBN, and in particular towards applying them to troubleshooting. Using nsDBNs is efficient, and facilitates modeling of complex structures and processes. Since the method is new, there are several interesting questions to investigate. For example, how inference should be made most efficiently, or how dynamics within an epoch can be modeled using stationary DBN. We will turn to these questions in our future work. The main driving force for developing the event driven nsDBN presented here is its application to decision theoretic troubleshooting, and by now we are satisfied with noting that with event driven

nsDBN, we can avoid assumptions, present in previous literature, on single faults, special structures of dependencies, or the use of function control nodes.

References

[Breese and Heckerman, 1996] Breese, J. S. and Heckerman, D. (1996). Decision-theoretic troubleshooting: A framework for repair and experiment. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*.

[Elliott et al., 2001] Elliott, R. J., Hunter, W. C., and Jamieson, B. M. (2001). Financial Signal Processing: A Self Calibrating Model. *International Journal of Theoretical & Applied Finance*, 4:567–584.

[Heckerman et al., 1995] Heckerman, D., Breese, J. S., and Rommelse, K. (1995). Decision-theoretic troubleshooting. *Communications of the ACM*, 38(3):49–57.

[Jensen and Nielsen, 2007] Jensen, F. V. and Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs*. Springer.

[Langseth and Jensen, 2002] Langseth, H. and Jensen, F. V. (2002). Decision theoretic troubleshooting of coherent systems. *Reliability Engineering & System Safety*, 80(1):49–62.

[Lauritzen, 1999] Lauritzen, S. L. (1999). Causal inference from graphical models. In *In Complex Stochastic Systems*, pages 63–107. Chapman and Hall/CRC Press.

[Mamon, 2002] Mamon, R. S. (2002). A time-varying Markov chain model of term structure. *Statistics & Probability Letters*, 60:309–312.

[Murphy, 2001] Murphy, K. (2001). The bayes net toolbox for matlab. *Computing Science and Statistics*, 33.

[Murphy, 2002] Murphy, K. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, UC Berkeley, USA.

[Neapolitan, 2003] Neapolitan, R. (2003). *Learning Bayesian Networks*. Prentice Hall.

[Olive et al., 2003] Olive, X., Trave-Massuyes, L., and Poulard, H. (2003). AO* variant methods for automatic generation of near-optimal diagnosis trees. In *14th International Workshop on Principles of Diagnosis (DX 03)*, pages 169–174.

[Pearl, 2000] Pearl, J. (2000). *Causality.* Cambridge.

[Pernestål et al., 2009] Pernestål, A., Warnquist, H., and Nyberg, M. (2009). Modeling and troubleshooting with interventions. In *Proceedings of 2nd IFAC Workshop on Dependable Control of Discrete Systems (DCDS 02).*

[Queen and Albers, 2009] Queen, C. M. and Albers, C. J. (2009). Intervention and Causality: Forecasting Traffic Flows Using a Dynamic Bayesian Network. *Journal of the American Statistical Association*, 104:669–681.

[Robinson and Hartemink, 2008] Robinson, J. W. and Hartemink, A. J. (2008). Non-stationary dynamic bayesian networks. In *Proceedings of NIPS.*

[Russell and Norvig, 2003] Russell, S. and Norvig, P. (2003). *Artificial Intelligence. A Modern Approach.* Prentice Hall.

[Skaanning et al., 2000] Skaanning, C., Jensen, F. V., and Kjærulff, U. (2000). Printer troubleshooting using bayesian networks. In *IEA/AIE '00: Proceedings of the 13th international conference on Industrial and engineering applications of artificial intelligence and expert systems*, pages 367–379.

[Spirtes et al., 2001] Spirtes, P., Glymour, C., and Scheines, R. (2001). *Causation, prediction, and Search. Second Edition.* MIT press.

[Warnquist and Nyberg, 2008] Warnquist, H. and Nyberg, M. (2008). A heuristic for near-optimal troubleshooting using ao^*. In *Proceedings of the 19th International Workshop on Principles of Diagnosis.*

[Warnquist et al., 2009] Warnquist, H., Pernestål, A., and Nyberg, M. (2009). Anytime near-optimal troubleshooting applied to an auxiliary truck braking system. In *Proceedings of 6th IFAC Symposium on Fault Detection, Supervision and Safety of technical processes (SAFEPROCESS 2009).*

Paper 4

Modeling and Efficient Inference for Troubleshooting Automotive Systems[1]

Anna Pernestål[*], Mattias Nyberg[*], and Håkan Warnquist[‡]

[*]*Division of Vehicular Systems, Department of Electrical Engineering, Linköping University, Sweden.*

[‡]*Artificial Intelligence & Integrated Computer Systems Division, Department of Computer and Information Sciences, Linköping University, Sweden*

Abstract

We consider computer assisted troubleshooting of automotive vehicles, where the objective is to repair the vehicle at as low expected cost as possible. The work has three main contributions: a troubleshooting method that applies to troubleshooting in real environments, the discussion on practical issues in modeling for troubleshooting, and the efficient probability computations. The work is based on a case study of an auxiliary braking system of a modern truck. We apply a decision theoretic approach, consisting of a planner and a diagnoser. Two main challenges in troubleshooting automotive vehicles are the need for disassembling the vehicle during troubleshooting to access parts to repair, and the difficulty to verify that the vehicle is fault free. These facts lead to that probabilities for faults and for future observations must be computed for a system that has been subject to external interventions that cause changes the dependency structure. The probability computations are further complicated due to the mixture of instantaneous and non-instantaneous dependencies. To compute the probabilities, we develop a method based on an algorithm, *updateBN*, that updates a static BN to account for the external interventions.

[1]This report is also available from Department of Electrical Engineering, Linköping University, S-58183 Linköping. LiTH-ISY-R-2921. It is partly based on [Pernestål et al., 2009] and [Warnquist et al., 2009].

1 Introduction

To meet increasing requirements on functionality, safety, and environmental performance, modern automotive vehicles become more and more complex products integrating electronics, mechanics and software. Due to their intricate architecture and functionality they are often difficult for a workshop mechanic to troubleshoot. In addition, shortened repair times and increased uptime are required. To shorten repair times for the increasingly complex automotive systems, one approach is to provide a computer aided troubleshooting system to the workshop mechanic. The troubleshooting system should to suggest a sequence of actions, including for example repairs and observations, that leads to a fault free vehicle at lowest expected cost.

One of the main driving factors in the current work is to design a troubleshooting system that is applicable to real automotive vehicles. The work is inspired by an application study of an auxiliary heavy truck breaking system, called the *retarder*. The retarder is a mechatronic system consisting of electric, mechanic, and hydraulic parts. In the work we discuss practical issues for modeling for troubleshooting, exemplified by modeling of the retarder.

In the literature, one approach to troubleshooting that has proven to be efficient the decision theoretic, see for example [Sun and Weld, 1993], [Heckerman et al., 1995], [Langseth and Jensen, 2002], and [Olive et al., 2003]. However, application studies in these previous works mainly consists of electronic systems, such as printers and electronic control units. In comparison with these electronic applications, the automotive mechatronic system considered here imply that the solution to the troubleshooting problem needs to take two important additional issues into account.

First, in automotive mechatronic systems it is often not as straightforward to determine whether a certain repair has made the system fault free as in the previous works. In the previous works, it is assumed that after each repair it is verified whether the system is fault free or not. Such a verification is typically expensive in automotive mechatronic systems, and therefore we do not presume it. The consequence is that we need to compute probabilities in a system subject to external interventions, i.e. after affecting the system with the repairs. Second, not all parts of the retarder can be reached without first disassembling other parts of the system. This means that the level of disassembly, and the extra time required for disassembly and assembly activities, needs to be considered in the solution.

During troubleshooting the aim is to guide the mechanic by suggesting the next repair or observation such that the expected repair cost is minimized. For small systems, the problem could be solved using influence diagrams [Jensen and Nielsen, 2007, Russell and Norvig, 2003]. For larger systems, such as troubleshooting of the retarder, influence diagrams becomes unfeasibly large and complex. Instead, we formulate the troubleshooting problem as a probabilistic

conditional planning problem.

The troubleshooter designed in this paper consists of a diagnoser and an action planner. The planner finds a conditional plan of actions by solving a general state space search problem, where a state describes the current knowledge, i.e. the current belief state, of the system. To achieve this plan, the planner has to consider the costs of actions and the effects they may have on the system and the of the system. The costs of actions are dependent on the level of disassembly, and each action may change this level. The planner asks the diagnoser for the effect that an action may have on the belief state and for the likelihoods of future action results. We use the heuristic search algorithm AO^* [Nilsson, 1980] to find an optimal plan, i.e. a plan with minimal expected cost that makes the vehicle fault free. The output from the planner to the mechanic is the first action of this plan. If the mechanic is busy waiting for a response, the search time contributes to the total repair cost. Therefore, the planner can be halted anytime returning a possibly suboptimal choice of action.

The diagnoser supports the planner with computation of probabilities of faults and of future observations. The main challenge in the probability computations are the external interventions, caused by the troubleshooting activities. These interventions change the structure of dependencies during the troubleshooting. The probability computations are further complicated by the two different kinds of dependencies, instant and non-instant, caused by the nature of the troubleshooting problem. In previous works on troubleshooting, computing probabilities after external interventions with the system is often avoided, for example by assuming a function-verifying observation after each repair, see for example [Langseth and Jensen, 2002]. In [Breese and Heckerman, 1996] interventions are handled using so called persistence nodes, where mapping nodes are used to track dependency changes. However, the dependency changes studied in the current paper are of a different source, and the persistence nodes are not applicable. Another approach is to utilize event-driven non-stationary dynamic Bayesian networks (event-driven nsDBN), see for example and [Pernestål and Nyberg, 2009]. In the event-driven nsDBN, new time slices are added to a dynamic Bayesian network (DBN) by events, caused by external interventions. By allowing different structures in different time slices the nsDBNs provides a general description of the troubleshooting process, but this generality complicates inference. In the current work, one step further is taken, and a new method for inference for troubleshooting is developed. We note that the probability computations in troubleshooting is of a special kind, and show how these probabilities can be computed by replacing the nsDBN with a static Bayesian network (BN) that is updated as troubleshooting progress.

To summarize, there are three main contributions in the current work: the complete troubleshooting strategy, the detailed investigation of practical issues when modeling and building troubleshooting systems for automotive vehicles,

and the new algorithm for efficient computation of the probabilities needed for troubleshooting.

We begin by introducing notation and preliminaries in Section 2, before presenting the retarder and troubleshooting scenario in Section 3. We describe the planner in Section 4, and in Section 5 we discuss modeling for troubleshooting. In particular we highlight practical issues when modeling real systems. The diagnoser is discussed in Section 6, and the BN updating algorithm used in the diagnoser is presented and proved in Section 7. Finally, we apply the troubleshooting system to the retarder in Section 8, before concluding and providing an outlook in Section 9.

2 Preliminaries

Before going into the troubleshooting details, we present the notation used, and give a brief introduction to Bayesian networks (BN) and dynamic Bayesian networks (DBN).

2.1 Notation

All variables considered in this work are discrete. We use capital letters for variables and lower case letters for their values, $X = x$. We use $p(X = x)$ or $p(x)$ to denote the probability that $X = x$, while $p(X)$ denotes the probability distribution of X. Bold face letters denote vectors. Subscripts are used to denote variable indicies, and superscripts to denote time. For example, x_i^t is the value of the variable X_i^t, with number i at time t.

2.2 Bayesian Networks

A Bayesian network (BN) is a directed acyclic graph representing a factorization of the joint probability distribution over a set $\{X_1, \ldots, X_n\}$ of variables. In the BN, denoted B, nodes represent variables[2] and edges between them represent dependency relations. We let $pa_B(X)$, $ch_B(X)$, and $de_B(X)$ denote the sets of parents, children, and descendants of variable X in the BN B. Moreover, we use $pa_B(x)$ to denote an assignment of values to $pa_B(X)$, and similarly for $ch_B(x)$ and $de_B(x)$. Whenever the BN B is clear from the context, we omit superscript B.

To each variable X_i in B, there is a conditional probability distribution (CPD) associated, defining the probability distribution $p(X_i|pa_B(X_i))$, and B

[2] we will use the terms "nodes" and "variables" interchangeable

represent the factorization

$$p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i|pa(X_i)). \quad (1)$$

We use the term evidence to denote assignments of values to variables in the BN. The BNs considered in this work are causal BNs, meaning that the direction of the edges represent causal effects. More detailed descriptions on BNs can for example be found in [Jensen and Nielsen, 2007, Russell and Norvig, 2003].

To model dynamic systems and processes, a dynamic Bayesian network (DBN) can be used. The DBN consists of time slices, where each time slice models the system at during certain time interval. Dependencies over time are represented by edges between the time slices, sometimes called temporal edges. For references on DBN, see for example [Jensen and Nielsen, 2007, Russell and Norvig, 2003, Murphy, 2002].

In a BN, two distinct set of nodes $\mathcal{A}$ and $\mathcal{B}$ are said to be d-separated [Jensen and Nielsen, 2007] given a third distinct set of nodes $\mathcal{U}$ if every path Π between nodes in $\mathcal{A}$ and $\mathcal{B}$ satisfies at least one of the following tree conditions: (a) Π contains a serial connection $\rightarrow U \rightarrow$ and $U \in \mathcal{U}$, (b) Π contains a diverging connection $\leftarrow U \rightarrow$ and $U \in \mathcal{U}$, or (c) Π contains a converging connection, $\rightarrow W \leftarrow$, such that neither W nor any descendant of W is in $\mathcal{U}$. That $\mathcal{A}$ and $\mathcal{B}$ are d-separated by $\mathcal{U}$ means that no information can flow from $\mathcal{A}$ to $\mathcal{B}$ when all nodes in $\mathcal{U}$ are assigned evidence. In probability computations, this means that $\mathcal{A}$ and $\mathcal{B}$ are conditionally independent given $\mathcal{U}$.

3 The Troubleshooting Scenario and System

In this section we present the troubleshooting scenario, and give an overview of the troubleshooting system, but we first present our motivating application: the retarder.

3.1 Motivating Application - the Retarder

The retarder is an auxiliary hydraulic braking system that allows braking of the truck without applying the conventional brakes. It consists of a mechanical system and a hydraulic system, and is controlled by an electronic control unit (ECU), see Figure 1. The retarder generates breaking torque by letting oil flow through a rotor driven by the propeller axle causing friction. The kinetic energy is thereby converted into thermal energy in the oil that is cooled off by the cooling system of the truck. At full effect and high rpm, the retarder can generate as much torque as the engine. We have chosen to study the retarder since it is a representative system of heavy duty trucks, and since it is difficult to troubleshoot due to its complexity.

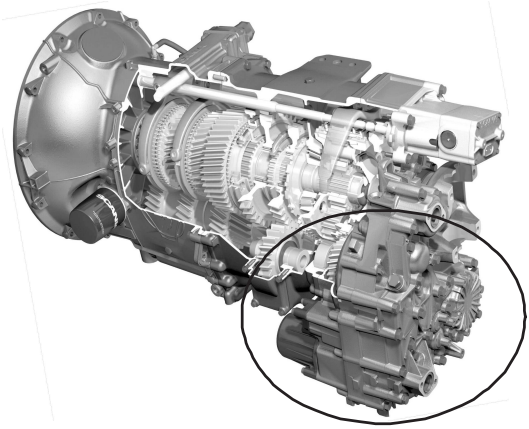


Figure 1: A heavy truck gearbox with an integrated retarder. The retarder is visible on the bottom right of the gearbox.

3.2 The Troubleshooting Scenario

Imagine a heavy truck, driving along the highway to deliver products to company. Suddenly, the driver experiences problems with the braking performance, and decides to take the vehicle to the workshop. At arrival to the workshop, the driver explains the problem to the mechanic, who plugs in his computer and reads out further information from the truck. From this information it is decided that the truck needs to be repaired immediately to avoid serious trouble. The driver must fulfill his transportation assignment, so the repair must be made as fast and time efficient as possible, and therefore the mechanic uses the computer aided troubleshooting system.

The troubleshooting system is connected to the truck, and suggests actions for the mechanic to perform. The mechanic reports the results to the troubleshooting system, and waits for new actions to be computed. This goes on until the troubleshooting system has declared that the truck can leave the workshop.

3.3 The Troubleshooting System

A troubleshooting action is a variable defined by its cost and its effect. The cost of an action is typically related to the time it takes to perform it and the resources consumed such as spare parts. Also, the cost depends on whether certain parts of the vehicle are assembled or not. The level of assembly is described by the *assembly state*. For example to replace the oil pressure sensor, the retarder oil needs to be drained and the oil cooler needs to be removed. The

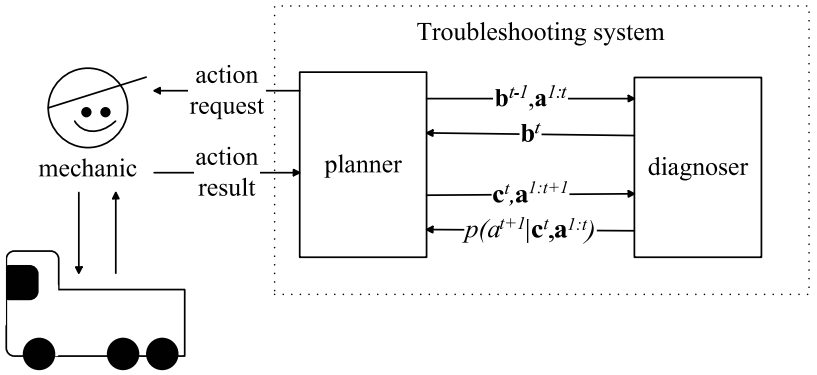


Figure 2: Overview of the troubleshooting system.

effect of an action can be to observe a value, perform a repair, test the operation of the truck, or to change the assembly state. When an action is performed we get an *action result*. An action result is a confirmed effect, i.e. the action and the outcome.

In Figure 2 an overview of the troubleshooting system used in this work is shown. The troubleshooting system communicates with the mechanic through *action requests* and the mechanic returns action results. There is no requirement that the result matches the request; the mechanic is free to perform activities on his own choice and report to the troubleshooting system. However, we presume that the mechanic is honest and only reports action results that actually have occurred.

As depicted in Figure 2, the troubleshooting system consists of two modules, a *planner* and a *diagnoser*, that communicate through the probabilities. This architecture divides the troubleshooting system into two parts with different tasks, and that can be developed independently.

To determine the next action the planner creates a conditional plan of actions called a *troubleshooting strategy*. This is done by searching the belief state space, i.e. the probability distribution

$$\mathbf{b}^t = p(\mathbf{C}^t|\mathbf{a}^{1:t}),$$

over component states $\mathbf{C}^t$, given the action results, $\mathbf{a}^{1:t} = \langle a^1, \ldots, a^t \rangle$, reachable from the current belief state. As shown in Figure 2, the planner utilizes the diagnoser in two ways: to compute the belief state $\mathbf{b}^t$ from the previous belief state $\mathbf{b}^{t-1}$ and the sequence $\mathbf{a}^{1:t}$ of action results, and to determine the probability $p(a^{t+1}|\mathbf{c}^t, \mathbf{a}^{1:t})$ of future actions. In the diagnoser, the probability computations are divided into two subproblems:

- Model updating: for maintaining a model of the current system, taking external interventions into account.

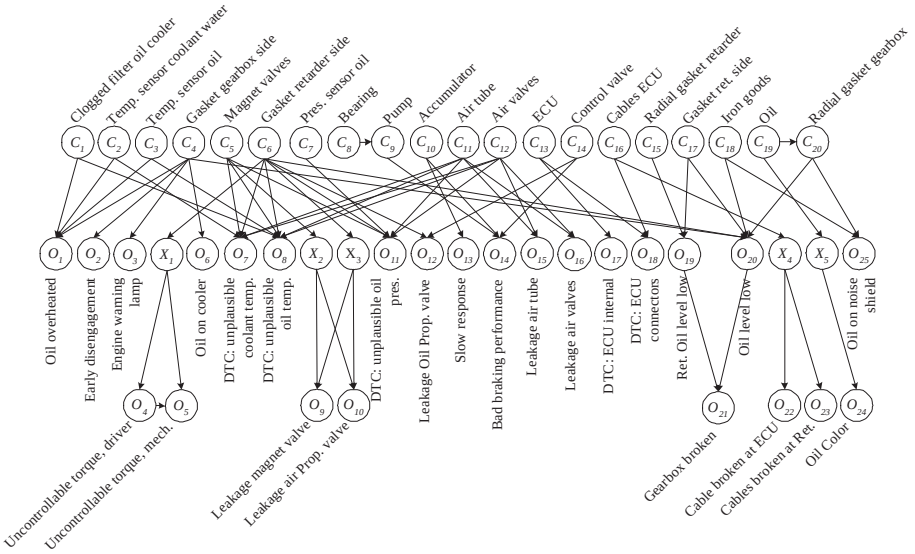


Figure 3: A Bayesian network modelling the retarder.

- Probability computation: for belief state updating and prediction of the outcomes of future actions.

Troubleshooting is terminated when the probability that the vehicle is fault-free is above a predefined threshold. Such a state is called a *goal state* for the planner.

3.4 Variables

We use a BN to model the system under troubleshooting in the diagnoser. The BN for the retarder is shown in Figure 3. As seen in the figure, there are three types of nodes: components, observable symptoms, and internal states. In this section we describe their characteristics.

Components

We use the term component both for the physical components and for the variables describing the status of the component. Components are denoted C_i, $i = 1, \ldots, N$. An assignment $\mathbf{C}^t = \mathbf{c}^t$ to all component variables is called a *diagnosis*. Each component C_i has the possible value "No Fault" (NF). In addition it has at least one fault state. To simplify the presentation we only consider one fault state, "Faulty" (F), for each component in the retarder.

There are two probability distributions related to the components. The first is the probability of a component being faulty given the operation history

H, $p(C_i|H)$. The history consists of information about how the vehicle has been used. For example, if the vehicle has been operated at extremely high load, its components are more likely to break. To avoid clutter in notation, we simply assume that $p(C_i|H) = p(C_i)$ in the current work. The second probability distribution for the components is the probability distribution of successful repair,

$$p(C_i|repair(C_i)). \tag{2}$$

We assume that, during troubleshooting, components can not change state spontaneously, i.e. if a component is faulty, it must be repaired in order to become fault free. The operation time during test drives is assumed to be short enough for no new faults to appear.

Observable Symptoms

Observable symptoms are represented by variables O_j, $j = 1, \ldots, M$, and represent observations that can be made, for example *Air leakage at Proportional valve* and *Engine warning lamp*. Observable symptoms are typically driver's observations, observations made in the workshop, Diagnostic Trouble Codes (DTC) generated in the ECU during driving, or direct observations of components. A direct observation is obtained by inspection of a component whether it is faulty or not.

When an observation action is confirmed, evidence is added to the corresponding observable symptom variable.

Internal States

In addition to the components and the observable symptoms, we use a set of hidden variables to represent internal states of the retarder. The internal states are represented by variables X_k, $k = 1, \ldots, L$. For example, in the retarder, there is an internal state representing the *Uncontrollable Braking Moment*. This internal state can be observed by both the mechanic and the driver. In this way we can model the fact that the result of observing the braking moment level may give different results for example due to the skill of the observer.

Troubleshooting BN

The three different types of variables presented above can be combined to a BN. In this work, we consider troubleshooting BNs defined as follows.

Definition 1 (Troubleshooting BN)**.** *A* Troubleshooting BN *consists of component variables, observable symptoms, and internal states, connected by directed edges such that the following rules hold:*

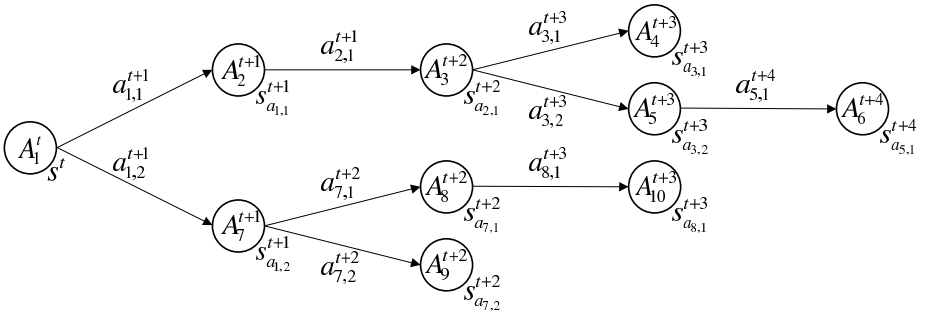


Figure 4: An example of a troubleshooting strategy described as a rooted tree.

- *Components can be parents to all kind of variables, but can only be children of other component variables.*
- *Observable symptoms can be parents only to other observable symptoms, but can be children of all types of variables.*
- *Internal states can be parents to observable symptoms only, and children to components only.*

4 Planner

As described in the previous section, the task of the planner is to generate the next action request A^{t+1}. This is done by evaluating different troubleshooting strategies and choosing the first action of the strategy with smallest expected cost. A troubleshooting strategy is a conditional plan which means that, depending on the results of previous actions, the following actions to take may be different. A troubleshooting strategy π is defined as a tree where each node is represented by an action A and each outgoing edge from a node represent an action result a of the corresponding action, see Figure 4. Branching occurs when an action has multiple possible results. The troubleshooting strategy is associated to a state s^t describing the vehicle at the time t when the strategy begins. This state consists of the assembly state $\mathbf{d}^t$, the belief state $\mathbf{b}^t$, and the history of action results $\mathbf{a}^{1:t}$. A troubleshooting strategy $\pi(s^t)$ is said to be complete if the execution of every action on the path from the root node to any leaf node leads to a goal state, i.e. a fault free vehicle.

4.1 Optimal Expected Cost of Repair

To evaluate complete troubleshooting strategies, the expected cost of repair (ECR) is computed. The expected cost of repair is the expected cost of reaching

any leaf node of the troubleshooting strategy. If the first action A^{t+1} of a troubleshooting strategy $\pi(s^t)$ is performed, there will be a certain action result a^{t+1} with the probability

$$p(a^{t+1}|\mathbf{a}^{1:t}) = \sum_{\mathbf{c}^t} p(a^{t+1}|\mathbf{c}^t, \mathbf{a}^{1:t}) \underbrace{p(\mathbf{c}^t|\mathbf{a}^{1:t})}_{\mathbf{b}^t}, \tag{3}$$

where we have marginalized over the component states $\mathbf{c}^t$ at time t. The first probability in the sum above is computed by the diagnoser, and the second is recognized as the previous belief state. Let the resulting state after action a^{t+1} be s_a^{t+1}, and let the cost of performing A^t be $cost(\mathbf{d}^t, A^t)$. Furthermore, let $\pi'(s_a^{t+1}) \subset \pi(s^t)$ be the troubleshooting strategy rooted in the node that is connected to by the edge corresponding to the action result a. Then, the expected cost of repair $\mathrm{ECR}(\pi(s^t))$ is

$$\begin{aligned}
&\mathrm{ECR}(\pi(s^t)) = \\
&= \begin{cases} cost(\mathbf{d}^t, A) & \text{if } s^t \text{ is goal state,} \\ cost(\mathbf{d}^t, A) + \sum\limits_{a^{t+1}} p(a^{t+1}|\mathbf{a}^{1:t})\mathrm{ECR}(\pi'(s_a^{t+1})) & \text{otherwise.} \end{cases}
\end{aligned}$$

For a given initial state s^t, the optimal troubleshooting strategy $\pi^*(s^t)$ is

$$\pi^*(s^t) = \arg\min_{\pi \in \Pi(s^t)} \mathrm{ECR}(\pi(s^t))$$

where $\Pi(s^t)$ is the set of all possible complete troubleshooting strategies starting in s^t. The optimal expected cost of repair $\mathrm{ECR}^*(s^t)$ is the expected cost of repair of $\pi^*(s^t)$. Let $\Pi_{A^{t+1}}(s^t)$ be the subset of $\Pi(s^t)$ where A^{t+1} is the first action. Then

$$\begin{aligned}
&\mathrm{ECR}^*(s^t) = \min_{\pi \in \Pi(s^t)} \mathrm{ECR}(\pi(s^t)) \\
&= \min_{A^{t+1}} \min_{\pi \in \Pi_A(s^t)} \mathrm{ECR}(\pi(s^t)) \\
&= \min_{A^{t+1}} \begin{cases} cost(\mathbf{d}^t, A) & \text{if } s^t \text{ is goal state,} \\ cost(\mathbf{d}^t, A) + \sum\limits_{a^{t+1}} p(a^{t+1}|\mathbf{a}^{1:t}) \min\limits_{\pi' \in \Pi(s_a^{t+1})} \mathrm{ECR}(\pi'(s_a^{t+1})) & \text{otherwise.} \end{cases} \\
&= \min_{A^{t+1}} \begin{cases} cost(\mathbf{d}^t, A) & \text{if } s^t \text{ is goal state,} \\ cost(\mathbf{d}^t, A) + \sum\limits_{a^{t+1}} p(a^{t+1}|\mathbf{a}^{1:t})\mathrm{ECR}^*(s_a^{t+1}) & \text{otherwise.} \end{cases}
\end{aligned} \tag{4}$$

4.2 Search Graph

To obtain the optimal troubleshooting strategy and the next action request, the minimization (4) must be solved. Figure 5 illustrates how the problem is decomposed in the form of a tree. Solving the minimization in (4) corresponds to

choosing to follow a single outgoing branch from the boxes in Figure 5. To compute the summation, every outgoing branch from the circles must be evaluated. This kind of decomposition corresponds to an AND/OR graph. In accordance with [Nilsson, 1980], the AND/OR graph can be defined as a hypergraph with nodes that are states interconnected by hyperedges. A hyperedge connects one state with one or many other successor states. In the AND/OR graph for (4), each non-goal state has one outgoing hyperedge for each action that connects to one other state for each action result of that action. A *solution* to an AND/OR graph is a subgraph of that graph that contains the start state and, for every non-goal state in the solution graph, exactly one hyperedge and all of its successor states. Every solution corresponds to a complete troubleshooting strategy and the optimal solution is the one that solves (4).

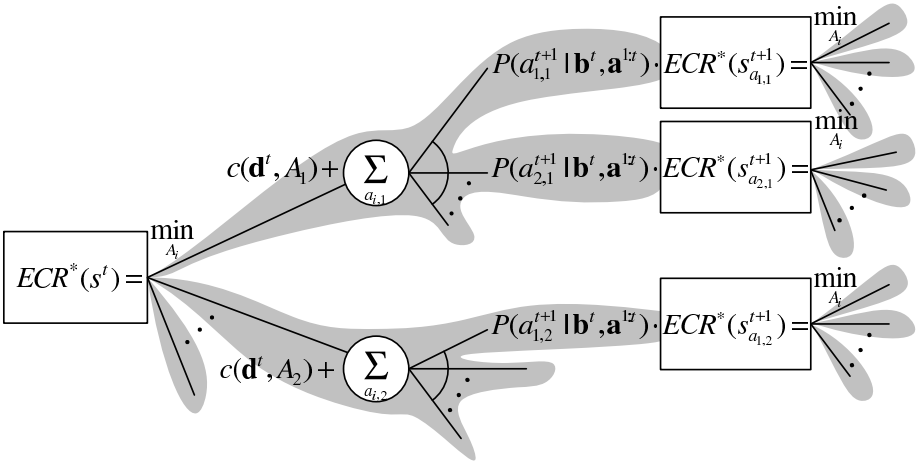


Figure 5: The problem decomposition of the calculation of (4).

Search Algorithm

There are many efficient algorithms to find optimal solutions in AND/OR graphs and the one used in this work is $AO*$ [Nilsson, 1980], [Martelli and Montanari, 1978]. $AO*$ is a heuristic search algorithm that finds the optimal solution to an implicit AND/OR graph Γ specified by a start state and a *successor function*. The successor function generates the successors s_a^{t+1} of a state s^t for every action result a^{t+1} as well as the probability of reaching each successor. The algorithm is initialized with an explicit AND/OR graph Γ' that consists of only the start state. It uses the successor function to expand Γ' with the successors of one of the leaf states in the optimal solution of Γ'. After each expansion

of Γ', the optimal solution is updated, i.e. the newly expanded state and all of its ancestors are evaluated using a cost function f. For the troubleshooting problem this is

$$f(s^t) = \min_{A^{t+1}} \begin{cases} cost(\mathbf{d}^t, A) & \text{if } s^t \text{ is goal state in } \Gamma, \\ h(s^t) & \text{if } s^t \text{ is leaf state in } \Gamma', \\ cost(\mathbf{d}^t, A) + \sum_{a^{t+1}} p(a^{t+1}|\mathbf{a}^{1:t}) f(s_a^{t+1}) & \text{otherwise,} \end{cases}$$

where $h(s^t)$ is a heuristic cost function that estimates the optimal expected cost of repair such that

$$h(s^t) \leq \mathrm{ECR}^*(s^t) \quad \text{for any state } s^t. \tag{5}$$

The algorithm keeps expanding Γ' until all leaf states in the optimal solution to Γ' are goal states. If (5) holds, then the optimal solution to Γ' is also the optimal solution to Γ.

The heuristic that is used in the implementation is derived from a relaxation of the problem, where we assume that we can observe all components for free. Then for all possible diagnoses $\mathbf{c}^t$, we have to compute the cost of repairing the faulty components in $\mathbf{c}^t$

$$h(s^t) = \sum_{\mathbf{c}^t} p(\mathbf{c}^t|\mathbf{a}^{1:t}) \sum_{\mathbf{C}_i^t = F} cost(\mathbf{d}^t, repair(C_i)). \tag{6}$$

where the probability $p(\mathbf{c}^t|\mathbf{a}^{1:t})$ can be taken directly from the belief state $\mathbf{b}^t$.

Finding the optimal solution to conditional planning problems is highly exponential [Rintanen, 2004]. This means that the time $AO*$ requires to complete can be very long. If the mechanic is waiting for a response, the computation time contributes to the cost. Therefore, the search can be aborted prematurely and the first action of the optimal solution to the current explicit graph Γ' is returned. This solution does not correspond to a complete troubleshooting strategy and the decision is therefore not necessarily optimal. However, for every additional computational time allowed, the quality of the solution converges monotonically toward the optimal.

5 Modeling for Troubleshooting

In this section we discuss modeling for troubleshooting. In particular, we discuss practical issues in modeling for troubleshooting, and we give an introduction to how event-driven non-stationary nsDBNs, deveolped in [Pernestål and Nyberg, 2009], can be used to handle external interventions during the troubleshooting process.

5.1 Practical Issues when Building BN for Troubleshooting

Building BNs for troubleshooting, as modeling in general, is an artwork that requires knowledge about the system to model and/or a lot of training data to learn the model from. Since troubleshooting support is most important when products are new, before experience is collected at the workshops, it is typically the case that the troubleshooting system, including the BN, should be available at the market at the same time as the vehicle is released. At this time, data is not yet collected, and the model must be learned mainly from expert knowledge.

The BN for the retarder shown in Figure 3 is based on engineers' expert knowledge, and consists of 20 component variables, denoted C_1 - C_{20}, five internal state variables, denoted X_1 - X_5, and 25 observable symptoms, denoted O_1 - O_{25}.

When building the BN we aim at a model that is simple enough to enable fast computations, but descriptive enough to solve the troubleshooting problem with sufficiently high precision. There are several design choices, and in this section we discuss some of the most important ones.

Components. The parts of the troubleshooted systems can be divided into components in the BN in different ways. The maximum size of components are sets of parts of the retarder that always are repaired together, also called *minimal repairable unit*. Choosing larger components may lead to that more parts than necessary are replaced during troubleshooting. Choosing smaller sets of parts of the retarder as components in the BN is possible, but may give worse performance in the troubleshooting algorithm and gives more parameters that need to be determined in CPDs.

In this work we choose components to be minimal repairable units. Furthermore, we allow several components to be faulty at the same time.

Driver or Mechanic. Observations concerning the performance of the vehicle, for example the braking torque, can be obtained by asking the driver or by letting the mechanic perform a test drive. In general, the answer from the mechanic is less uncertain but is often obtained at a higher cost since it is more expensive to let the mechanic perform a test drive than interviewing the driver. The driver's answers can only be obtained at the beginning of troubleshooting. It may be the case that the driver's answers bias the mechanic. For example, if the driver complains about uncontrollable braking torque it is reasonable that the mechanic will be influenced and observe the same symptom with higher probability. This case is modeled as a dependency between the observation nodes, see O_4 and O_5 in Figure 3 for an example.

Perception. In some observations there may be uncertainties. For example the observation *Leakage magnet valve* (O_{15}) can be mistaken for *Leakage prop. valve* (O_{16}). We model this by using internal state variables that represent the

true situation, in this case X_2 and X_3, and from each such internal variable to both observations.

Effect of External Systems. In the troubleshooting of a certain system, there are typically adjacent systems that also may affect the observations. One previously used approach is to assume that surrounding systems are fault free, see e.g. [Heckerman et al., 1995]. In the current work we take another approach. We consider two cases: when the troubleshooted system cause faults in an adjacent systems, and when faults in an adjacent system affects observations in the troubleshooted system.

In the first case, when the troubleshooted system cause a fault in an adjacent system, we model the adjacent system as an observation. In Figure 10 we have for example identified that the states *Retarder Oil Level Low* (O_{19}) and *Oil Level Low* (O_{20}), which can also be observed and thus is modeled as observable symptoms, that may cause the gearbox to brake. This is modeled through the observable symptom *Gearbox Broken* (O_{21}).

An example of the second case, that faults in adjacent systems also can explain observations in the troubleshooted system, is that leakages outside the retarder may cause the observation *DTC: Unplausible Oil Pressure* (O_{11}). We take care of this external fault by increasing the probability of false alarm for this DTC. Note this also induces that the requirement on the goal state must be changed, i.e. at some point we consider the system as fault free although there may be observations that have alarmed.

Time. There are two aspects of time in troubleshooting. First, "time is money", in the sense that there are costs associated with having the truck at the workshop. To model this, each action has a cost for performing the action. This cost is taken into account in the planner.

Second, time goes on while troubleshooting, and the system may change over time. In particular, the system changes with repairs and test operation. In the current work we consider troubleshooting as a discrete process, where time steps are taken when repair and operation actions are performed. The time interval between two such actions may be of different length, and we assume that the system is static during each interval. This assumption is reasonable, since the vehicle is at rest at the workshop, and there are basically no dynamics present.

5.2 Repairs, Operations, and Interventions

Assume that there is a BN modeling the system under troubleshooting. Performing observations simply mean adding evidence to the observed variables in the BN. Performing a repair of component C_i, means that the repaired component is fault free with probability given by (2). However, when performing a repair there is also an intervention with the system. To illustrate the effect of a repair, assume for a moment that repairs are always successful. Then, repairing

a component C_i means that the component is forced to be fault free by intervention, rather than being observed as fault free. Therefore, it is not sufficient to only add the evidence $C_i = NF$ to the BN [Pearl, 2000]. The consequences of an external intervention depend on the characteristics of the causal dependencies in the system. In troubleshooting, there are two different kinds of causal relations: instant and non-instant. For example, if the oil is replaced in the retarder, this will have an instant effect on the oil color. Non-instant relations, on the other hand, need operation of the system to be present. One example is that if a gasket is replaced in the retarder, the retarder must be operated in order to verify if there is a leakage or not. In this small example it is shown that operation actions also are external interventions with the systems since an operation changes the relations between variables.

The nature of interventions and their causal effects is carefully discussed by Pearl in [Pearl, 2000]. However, the interventions considered in [Pearl, 2000] are based on that all causal dependencies are instant, i.e. that changing the value of a variable gives instantaneous effects on its children. In the troubleshooting application there are both instant and non-instant causal dependencies, and thus the rules of causality developed in [Pearl, 2000] are not directly applicable.

5.3 Event-Driven Non-stationary DBN

To compute probabilities of faults after external interventions, i.e. after repairs and operations, a model describing both the system under troubleshooting and the troubleshooting process itself is needed. One framework for modeling troubleshooting processes is the one based on event-driven non-stationary DBN (event-driven nsDBN) developed in [Pernestål and Nyberg, 2009]. An nsDBN is a DBN, where dependencies are allowed to be different in different time slices, see for example [Robinson and Hartemink, 2008, Pernestål and Nyberg, 2009]. In an event-driven nsDBN, new time slices are generated by external interventions that change the structure of dependencies. Following the nomenclature in [Pernestål and Nyberg, 2009], such external interventions are called *events*. An example of an event-driven nsDBN is shown in Figure 6. A time interval between two events is called an *epoch*. As discussed in Section 5.1, we assume that the system is static between events, meaning that in the nsDBN, each epoch is modeled by a time slice. In an epoch several observations can be performed. However, we assume that the same observable symptom can only be observed once in each epoch. An nsDBN together with a sequence of action results is called a *troubleshooting session*.

To get familiar with nsDBNs, study the example in Figure 6. The figure shows a three-time-slice nsDBN modeling a subsystem of the retarder. In the figure, subscripts correspond to numbers in Figure 3, and superscripts denote the corresponding time slice (or, equally, epoch). The nsDBN in Figure 6 has three time slices. The first models the system at arrival to the workshop. The

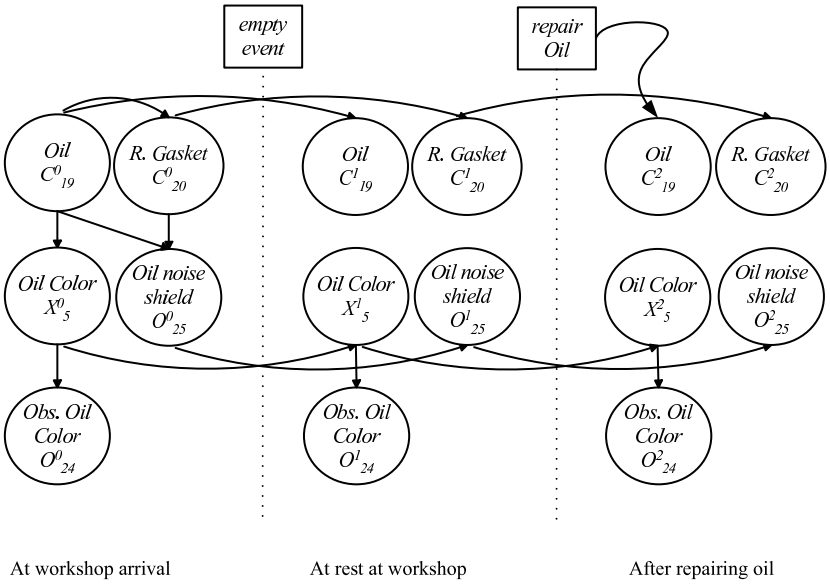


Figure 6: Dependencies in the a subsystem of the retarder at workshop arrival, at rest at the workshop, and after repairing the oil.

second time slice is started by the "empty event", i.e. the event where there is no external intervention with the system. The system is at rest at the workshop, and no actions have been performed. As described in [Pernestål and Nyberg, 2009], the empty event is merely for theoretical purposes where it is used as a reference; in practice, there is no need for starting a new epoch after the empty event, since the system has not changed. The third and final time slice in Figure 6 is initialized by the event that the oil has been repaired. Using the nsDBN in Figure 6, reasoning during troubleshooting can proceed in the following way. In the figure, ignoring the directions of the edges, there is a path between O^1_{22} and C^1_{19} (via O^0_{22}, C^0_{20}, and C^0_{19}). This means that by observing whether there is oil on the noise shield, conclusions can be drawn about the status of the oil. In the third time slice, after repairing the oil, the path from O^2_{22} and C^2_{19} is broken, and the observation whether there is oil on the noise shield does not contribute in the reasoning about the state of the oil.

In each time slice in an nsDBN, there are two types of edges: instant edges and non-instant edges. We use the following definition from [Pernestål and Nyberg, 2009], slightly rewritten to fit into the current framework.

Definition 2 (Instant Edge)**.** *An edge in a BN that models a system is* instant *if it does not require operation of the system to be present. An edge that not requires operation to be present is* non-instant.

In Figure 6, the edge between the *Oil* and *Oil Color* is instant, while the edge between *Radial Gasket* and *Oil on Noise Shield* is non-instant. Also, the nodes in an nsDBN can be classified as one of two types: persistent or non-persistent. Again, we use the definition from [Pernestål and Nyberg, 2009].

Definition 3 (Persistent Variable)**.** *A variable in a BN is* persistent *if its value in one time slice, generated by the empty event, is dependent on its value in the previous epoch. A variable that is not persistent is* non-persistent*.*

In Figure 6, the nodes *Oil*, *Radial Gasket*, *Oil Color*, and *Oil on Noise Shield* are persistent, while the node *Obs. Oil Color* is non-persistent. In particular, for a persistent variable, if there are no external interventions affecting it, there is an edge between the two copies of the variable in two consecutive time slices.

In [Pernestål and Nyberg, 2009] it is shown that an nsDBN modeling a troubleshooting process can be characterized by three pieces of information: (i) an initial BN B^0_{ns}; (ii) the effects of the empty event; and (iii) for each action, information about the edges added and removed, and the CPDs changed in relation to the effects of the empty event.

We use the following assumptions on related to the nsDBN.

Assumption 1 (Initial BN)**.** *The initial BN B^0_{ns} is a troubleshooting BN as defined by Definition 1.*

Assumption 2 (Persistence)**.** *If not affected by external interventions, a persistent variable has the same value in two consecutive epochs.*

Assumption 3 (Persistent Components)**.** *Components are persistent.*

Assumption 4 (Empty Event)**.** *The empty event in epoch t generates a new time slice B^{t+1}_{ns} where all nodes and all instant edges are copied from the previous time slice B^t_{ns}. Time slice B^{t+1}_{ns} is connected to B^t_{ns} by edges from all persistent variables in B^t_{ns} to its copies in B^{t+1}_{ns}.*

Assumption 5 (Locality of Repair)**.** *The event repair(C_i) in epoch t generates a new time slice B^{t+1}_{ns} that is equal to the time slice generated by the empty event, except that the edge between C^t_i in B^t_{ns} and C^{t+1}_i in B^{t+1}_{ns} is removed. In addition, all edges between C_i in B^{t+1}_{ns} and all other components in B^{t+1}_{ns} are removed.*

Assumption 6 (Operation)**.** *The event operate in epoch t generates a new time slice B^{t+1}_{ns} that is equal to the initial time slice B^0_{ns}. Time slice B^{t+1}_{ns} is connected to B^t_{ns} by edges from each component variable in B^t_{ns} to its copy in B^{t+1}_{ns}.*

One consequence of the assumptions above is that, with only one exception, no faults are introduced during troubleshooting. The exception is that the

repair of a component C_i may be unsuccessful, and introduce faults in C_i. Moreover, Assumption 5 means that repair of a component C_i does not affect any other components than C_i. Assumption 6 means that operation during troubleshooting is long enough for all non-instant dependencies to establish. Furthermore, it means that test operation makes all persistent variables, except components, independent of their previous values given the current component states.

6 Diagnoser: Belief State Updating

In this section and the following, we present the computations performed in the diagnoser. As described in Section 3.3 the computations are divided into two subproblems. The first subproblem, to maintain a model of the troubleshooted system, is considered in Section 7, and in this section we concentrate on the second subproblem: probability computations for belief state updating and for prediction of future observations.

As described in Section 3.3, there are two cases where the planner requests probabilities from the diagnoser. The first case is when an action result a^t is reported to the planner, and planner requests the diagnoser to compute the belief state, i.e. the probability distribution

$$\mathbf{b}^t = \mathbf{b}(\mathbf{c}^t) = p(\mathbf{c}^t|\mathbf{a}^{1:t}), \tag{7}$$

for $\mathbf{c}^t = (c_1^t, \ldots, c_N^t)$, given a sequence $\mathbf{a}^{1:t} = \langle a^1, \ldots, a^t \rangle$ of action results. Recall also that the previous belief state is known, although not explicitly written in the probabilities. The second case is during planning, and regards the probability distributions of possible future actions, $p(a^{t+1}|\mathbf{c}^t, \mathbf{a}^{1:t})$, i.e. the first probability in the sum in (3). Repair and observation actions are requests to the mechanic to perform an activity, and have only one possible result each, namely "repair performed" or "operation performed". These action results are always obtained with probability one. For observations, on the other hand, there are several different values on the observed variable. Therefore, the diagnoser needs to compute the first probabiltiy probability

$$p(o_j^{t+1}|\mathbf{c}^t, \mathbf{a}^{1:t}). \tag{8}$$

This probability will be computed in Section 7. The remainder of this section is devoted to computation of the belief state (7) for observation, repair, and operation actions. In the diagnoser, there is no need to consider assemble/disassemble actions since they do not introduce any new faults, and thus do not change the belief state.

6.1 Observation Actions

Let $a^t = observe(O_j = o_j)$. By Assumption 2 we have that

$$p(\mathbf{C}^t = \mathbf{c} | \mathbf{C}^{t-1} = \mathbf{c}, \mathbf{a}^{1:t}) = 1, \tag{9}$$

and, by using Bayes' rule, (7) can be written as

$$p(\mathbf{c}^t | \mathbf{a}^{1:t}) = p(\mathbf{c}^t | o_j^t, \mathbf{a}^{1:t-1}) = \gamma p(o_j^t) | \mathbf{c}^{t-1}, \mathbf{a}^{1:t-1}) \underbrace{p(\mathbf{c}^{t-1} | \mathbf{a}^{1:t-1})}_{\mathbf{b}^{t-1}}, \tag{10}$$

where γ is a normalization constant, and we have used (9) to replace $\mathbf{c}^t$ with $\mathbf{c}^{t-1}$ in the last equality. In (10), the previous belief state $\mathbf{b}^{t-1}$ is known, so the resulting probability computation to perform is

$$p(o_j^t | \mathbf{c}^{t-1}, \mathbf{a}^{1:t-1}), \tag{11}$$

which is of the same form as (8), and will be computed in Section 7.

6.2 Repair Actions

Let $a^t = repair(C_i)$, let $s \in \{NF, F\}$ and $\mathbf{C}_{\bar{i}} = (C_1, \ldots, C_{i-1}, C_{i+1}, \ldots, C_N)$. The belief state after repairing C_i at time t is then

$$\begin{aligned}
&\mathbf{b}(c_1^t, \ldots, c_{i-1}^t, s, c_{i+1}^t, \ldots, c_N^t) = \\
&= p(\mathbf{c}_{\bar{i}}^t, C_i^t = s | repair(C_i), \mathbf{a}^{1:t-1}) = \\
&= p(\mathbf{c}_{\bar{i}}^t | repair(C_i), \mathbf{a}^{1:t-1}) p(C_i^t = s | repair(C_i), \mathbf{a}^{1:t-1}) = \\
&= p(\mathbf{c}_{\bar{i}}^t | \mathbf{a}^{1:t-1}) p(C_i^t = s | repair(C_i)),
\end{aligned} \tag{12}$$

where we, in the second equality, have used that the repair makes $\mathbf{C}_{\bar{i}}^t$ and C_i^t independent. In the last equality of (12) we have used that $\mathbf{C}_{\bar{i}}^t$ is independent of the repair of C_i, and that, given that it is repaired, C_i^t is independent of previous events. Marginalizing over C_i^{t-1}, (12) becomes

$$\begin{aligned}
&p(\mathbf{c}_{\bar{i}}^{t-1} | \mathbf{a}_{1:t-1}) p(C_i^t = s | repair(C_i)) = \\
&= \big(p(\mathbf{c}_{\bar{i}}^{t-1}, C_i^{t-1} = NF | \mathbf{a}_{1:t-1}) + p(\mathbf{c}_{\bar{i}}^{t-1}, C_i^{t-1} = F | \mathbf{a}_{1:t-1})\big) \times \ldots \\
&p(C_i = s | repair(C_i)) = \\
&= \big(\mathbf{b}(\ldots, c_{i-1}^{t-1}, NF, c_{i+1}^{t-1}, \ldots) + \mathbf{b}(\ldots, c_{i-1}^{t-1}, F, c_{i+1}^{t-1}, \ldots)\big) p(C_i = s | repair(C_i)),
\end{aligned} \tag{13}$$

and belief state updating after $repair(C_i)$ is given by (13). Given the previous belief state $\mathbf{b}^{t-1}$ belief state updating after repair is simply an addition and a multiplication. In particular, under the assumption that repairs are always

successful, the updated belief state after a repair action becomes

$$\begin{aligned}&\mathbf{b}^t(\ldots, c_{i-1}, s, c_{i+1}, \ldots) = \\ &= \begin{cases} 0 & \text{if } s = F, \\ \mathbf{b}^{t-1}(\ldots, c_{i-1}^{t-1}, NF, c_{i+1}^{t-1}, \ldots) + \mathbf{b}^{t-1}(\ldots, c_{i-1}^{t-1}, F, c_{i+1}^{t-1}, \ldots) & \text{if } s = NF. \end{cases}\end{aligned} \tag{14}$$

6.3 Operation Actions

Finally, let $a^t = \mathit{operate}$. According to Assumption 6 no new faults appear during operation, and the belief state updating becomes

$$\mathbf{b}(\mathbf{c}^t) = p(\mathbf{c}^t|\mathbf{a}^{1:t-1}, \mathit{operate}) = p(\mathbf{c}^{t-1}|\mathbf{a}^{1:t-1}) = \mathbf{b}(\mathbf{c}^{t-1}). \tag{15}$$

7 Diagnoser: BN Updating

In the previous section, it is shown that for repair and operation actions the belief state is simply updated from the previous belief state according to (13) and (15). For observation actions, probabilities of the type (11) are needed to update the belief state. This probability is the same as (8). It can not be obtained by simple manipulations of the previous belief state only, and needs to be computed in the diagnoser. One straight-forward approach to compute the probability (11) is to use an event-driven nsDBN as described in Section 5.3. The event-driven nsDBN is a general model of the troubleshooting process, but, due to its generality, probability computations in an event-driven nsDBN may become time consuming and inefficient. In this section we will take off from the framework of event-driven nsDBN and develop an algorithm that efficiently updates a static BN instead of unrolling an nsDBN. To apply the nsDBN we begin with dividing the sequence $\mathbf{a}^{1:t}$ of actions into two sequences, one comprising the events, $\mathbf{e}^{1:t}$, and one comprising the evidence, $\mathbf{v}^{1:t}$. For example,

$$\begin{aligned}&\mathbf{a}^{1:3} = \langle \mathit{repair}(C_1), \mathit{observe}(O_2 = o_2), \mathit{operate}\rangle \quad \text{gives} \\ &\mathbf{e}^{1:3} = \langle \mathit{repair}(C_1), 0, \mathit{operate}\rangle, \\ &\mathbf{v}^{1:3} = \langle 0, \mathit{observe}(O_2 = o_2), 0\rangle.\end{aligned}$$

Above, the figure '0' is used to denote that there is no event or evidence respectively. Let $B_{ns}(\mathbf{e}^{1:t})$ be the nsDBN generated by the sequence $\mathbf{e}^{1:t}$ of events. The probability (11) can then be written

$$p(o_j^t|\mathbf{c}^{t-1}, \mathbf{a}^{1:t-1}) = p(o_j^t|\mathbf{c}^{t-1}, \mathbf{v}^{1:t-1}, B_{ns}(\mathbf{e}^{1:t-1})). \tag{16}$$

In the following we will sometimes write B_{ns} instead of $B_{ns}(\mathbf{e}^{1:t-1})$ if it is clear from the context which sequence of events that have generated the nsDBN.

Although the use of nsDBN is straight-forward, it may lead to too complex computations causing too long waiting times for the mechanic in many troubleshooting applications. As discussed in Section 3.2 the mechanic waits for the troubleshooting system to suggest new activities, and waiting times of no more than a few seconds are acceptable. The diagnoser is called several times for each step meaning that the computation time in the diagnoser has large impact on the waiting time.

During search in the planner, there are many sequences of actions under consideration at the same time, and the planner switches back and forth between these sequences. Each sequence of actions generates an nsDBN. There are two main alternatives for using nsDBNs for the probability computations. In the first alternative, no nsDBN is stored. Each time the planner switches to a new sequence of actions, all time slices of the nsDBN representing this sequence are unrolled and the probability computations are performed from start to the current time. In stationary DBNs the probability computations can be made efficiently using algorithms presented for example in [Murphy, 2002]. For nsDBNs, on the other hand, the structure changes lead to that these efficient methods can not be applied. Instead, basic inference methods such as variable elimination are applied [Jensen and Nielsen, 2007, Pernestål and Nyberg, 2009]. This may lead to time consuming computations in the nsDBN. The second alternative is, instead of generating a new nsDBN for each action sequence, to store one nsDBN for each action sequence. The nsDBN can be stored as the distribution of the variables in time slice $t-1$ together with the last two time slices. When (if) the planner returns to this particular sequence, a new epoch is added and for example variable elimination can be used to compute the new probabilities. Since the number of considered action sequences may be large, this approach may require an unfeasible memory capacity. Furthermore, if K is the number of nodes, inference is made in a BN with $2K$ nodes.

Taking another look at (16), we note that instead of using an nsDBN that can be used to compute arbitrary probabilities, it is sufficient to use a model that gives the conditional probabilities for the observations only. This opens the possibility to use a simpler model that is optimized for computation of the probabilities (16). The strategy here is to use a sequence of static BNs $B^0, B^1, \ldots$ such that

$$p(o_j^{(t)}|\mathbf{c}^{(t)}, \mathbf{v}^{1:t}, B_{ns}(\mathbf{e}^{1:t})) = p(O_j = o_j^{(t)}|\mathbf{C} = \mathbf{c}^{(t)}, \mathbf{v}^{1:t}, B^t). \tag{17}$$

The probability in the right hand side of (17) is computed in the static BN B^t, and we have introduced the convention that variables in the static BN have no superscript, but are assumed to belong to the BN that the probability is conditioned on. Moreover, recall that superscript on variables in an nsDBN denote the time slice they belong to. In (17) we have introduced superscript (t) to denote the time slice after event e^t but before next non-empty event. For example, let $a^t = \mathit{repair}(C_i)$, $a^{t+1} = \mathit{observe}(O_j = o_j)$, and $a^{t+2} = \mathit{observe}(O_l = o_l)$.

Then, since the observations are not events, we have that

$$p(a^{t+1}|\mathbf{c}^{(t)}, \mathbf{v}^{1:t}, B_{ns}(\mathbf{e}^{1:t})) = p(o_j^{(t)}|\mathbf{c}^{(t)}, \mathbf{v}^{1:t}, B_{ns}(\mathbf{e}^{1:t})),$$
$$p(a^{t+2}|\mathbf{c}^{(t)}, \mathbf{v}^{1:t}, B_{ns}(\mathbf{e}^{1:t})) = p(o_l^{(t)}|\mathbf{c}^{(t)}, \mathbf{v}^{1:t}, B_{ns}(\mathbf{e}^{1:t})).$$

For each sequence of action results under consideration in the planner, the belief state is stored, but no BNs are stored. Instead, when the planner switches to an action sequence $\mathbf{a}^{1:t}$, the BN B^t is generated from this sequence, and inference about future observations is performed in this BN. It will be shown that it is sufficient to perform inference in a subpart of B^t, typically consisting of a number of nodes that is significantly smaller than K.

7.1 BN Updating Example

We now present an algorithm $B^t = updateBN(B^{t-1}, a^t, \mathbf{b}^{t-1})$ that recursively generates the sequence of BNs $B^0, B^1, \ldots$ so that (17) is satisfied. To illustrate the idea of algorithm *updateBN*, consider again the example system with the two components *Oil* and *RadialGasket* introduced in Section 5.3. In Section 5.3 the nodes are classified as persistent or not, and the edges within time slices are classified as instant or not. Figure 7(a) shows an nsDBN modeling a troubleshooting process with two events (external interventions): *repair*(*Oil*) and *operate*. In the figure, non-instant edges are marked with dotted arrows while instant edges are solid. Persistent nodes are gray and non-persistent nodes are white.

The leftmost part of Figure 7(a), the time slice for epoch 0, or simply "time slice (0)", models the system when troubleshooting is initialized. In this time slice, nodes are marked with superscript (0) and is the initial BN, denoted $B_{ns}^0 = B_{ns}^{(0)}$ of the nsDBN. Below time slice (0), in Figure 7(b), the corresponding B^0 is shown. Since there has been no external interventions with the system, B^0 is identical to B_{ns}^0.

Updating Example: Repair

Let $a^1 = e^1 = repair(C_{19})$, i.e. that the oil is repaired. In the nsDBN in Figure 7(a) the event initializes epoch 1 and produces a new time slice. The new time slice is constructed by copying all nodes and instant edges from the previous time slice. According to Assumption 5 temporal edges are added between all persistent nodes, except between $C_{19}^{(0)}$ and $C_{19}^{(1)}$, which represents the oil before and after the repair. Since all probability queries will be of the type (11), we study how to compute the probabilities for the observations.

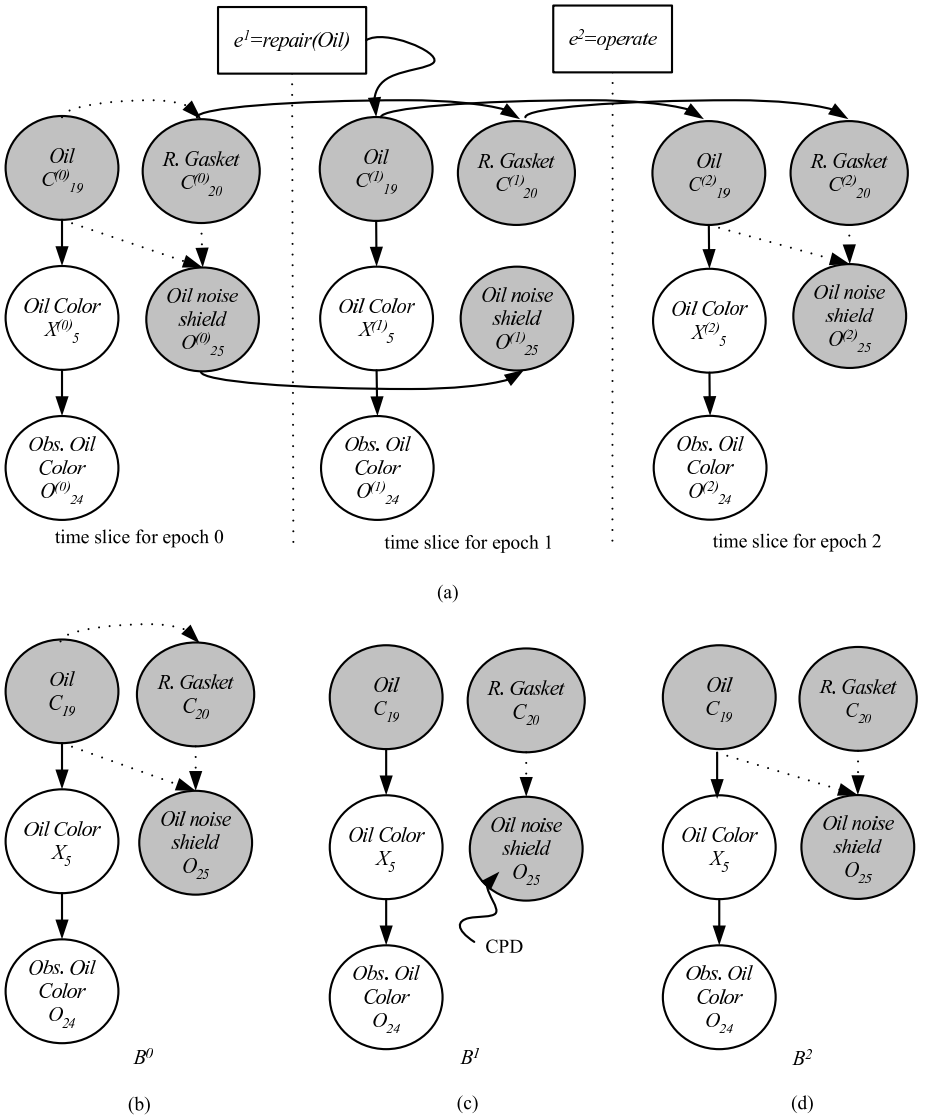


Figure 7: An nsDBN modeling the example system subject to a troubleshooting sequence (top), and the corresponding BNs (bottom).

Consider first the probability of $o_{24}^{(1)}$,

$$p(o_{24}^{(1)}|c_{19}^{(1)}, c_{20}^{(1)}, repair(C_{19})) = p(o_{24}^{(1)}|c_{19}^{(1)}, B_{ns}(\mathbf{e}^{1:1})) = \\ = \sum_{x_5^{(1)}} p(o_{24}^{(1)}|x_5^{(1)}, B_{ns}(\mathbf{e}^{1:1}))p(x_5^{(1)}|\mathbf{c}_{19}^{(1)}, B_{ns}(\mathbf{e}^{1:1})). \quad (18)$$

In the first in equality we have used (16) and that $o_{24}^{(1)}$ is independent of $c_{20}^{(1)}$ in $B_{ns}(\mathbf{e}^{1:1})$ and in the last equality we have marginalized over the internal variable $X_5^{(1)}$. The sum in (18) contains variables in time slice (1) only. Thus, the computations are independent of the variables in time slice (0).

Consider now the probability of $o_{25}^{(1)}$. Since the variables $O_{25}^{(1)}$ and $C_{20}^{(1)}$ are persistent and connected to their copies in the previous time slice, they have the same values in the two time slices and can be used interchangeably. By noting that $O_{25}^{(1)}$ is independent of $C_{19}^{(1)}$, and by marginalizing over $C_{19}^{(0)}$ we obtain

$$p(o_{25}^{(1)}|c_{19}^{(1)}, c_{20}^{(1)}, repair(C_{19})) = p(o_{25}^{(1)}|c_{20}^{(1)}, B_{ns}(\mathbf{e}^{1:1})) = p(o_{25}^{(0)}|c_{20}^{(0)}, B_{ns}(\mathbf{e}^{1:1})) = \\ = \sum_{c_{19}^{(0)}} p(o_{25}^{(0)}|c_{19}^{(0)}, c_{20}^{(0)}, B_{ns}(\mathbf{e}^{1:1}))p(c_{19}^{(0)}|c_{20}^{(0)}, B_{ns}(\mathbf{e}^{1:1})). \quad (19)$$

The last probability in the sum in (19) can be written as

$$p(c_{19}^{(0)}|c_{20}^{(0)}, B_{ns}(\mathbf{e}^{1:1})) = \frac{p(c_{19}^{(0)}, c_{20}^{(0)}|B_{ns})}{p(c_{20}^{(0)}|B_{ns}(\mathbf{e}^{1:1}))} = \frac{p(c_{19}^{(0)}, c_{20}^{(0)}|B_{ns}(\mathbf{e}^{1:1}))}{\sum_{c_{19}^{(0)}} p(c_{20}^{(0)}, c_{20}^{(0)}|B_{ns}(\mathbf{e}^{1:1}))}. \quad (20)$$

Here, $p(c_{20}^{(0)}, c_{20}^{(0)}|B_{ns}(\mathbf{e}^{1:1})) = \mathbf{b}^0$ is known, and will not change. Therefore, we can update the CPD $p(o_{25}^{(1)}|c_{20}^{(1)}, B_{ns}(\mathbf{e}^{1:1}))$ by using (19) and (20), and then forget the previous time slice.

The computations above show that, if the CPD for $O_{25}^{(1)}$ is updated, it is possible to compute the probabilities

$$p(o_{24}^{(1)}|\mathbf{c}^{(1)}, repair(C_{19})) \text{ and } p(o_{25}^{(1)}|\mathbf{c}^{(1)}, repair(C_{19}))$$

using variables in time slice (1) only. This indicates that, beginning with a BN B^0 corresponding to epoch 0, we can apply a sequence of manipulations on nodes and edges and obtain a new BN B^1 that corresponds to epoch 1. The two BNs B^0 and B^1 are shown in Figure 7(b) and (c) respectively. These manipulations are illustrated in Figure 8. They begin with an nsDBN consisting of the two epochs 0 and 1. First, we merge the nodes with the same values and remove superscript, i.e. $O_{25}^{(0)}$ and $O_{25}^{(1)}$ are merged to O_{25} and $C_{20}^{(0)}$ and $C_{20}^{(1)}$ are merged to C_{20} in Figure 8(b). In (20) it is shown that the probability for $C_{19}^{(0)}$ can be computed from $\mathbf{b}^0$. If the variables $X_5^{(0)}$ and $O_{24}^{(0)}$ have evidence this is taken into account in $\mathbf{b}^0$, and if they do not have evidence they are barren

nodes, see for example [Jensen and Nielsen, 2007], and will not contribute in the probability computations. Thus, $X_5^{(0)}$ and $O_{24}^{(0)}$ can be removed. This is illustrated by crossing over in the nodes in Figure 8(b). Finally, by updating the CPD for O_{25} according to (20) we can remove $C_{19}^{(0)}$ and obtain the BN B^1 in Figure 8(c).

Finally, we summarize the set of manipulations made on B^0 in Figure 7(b) to obtain B^1 in Figure 7(c).

- Set $B^1 = B^0$.
- Remove all non-instant edges to and from the repaired component C_{19}.
- Update the CPD for $O_{25} = O_{25}^{(2)}$ according to (20).

Updating Example: Operation

After repairing the oil, the system is operated, i.e. $a^2 = e^2 = \textit{operate}$. In the nsDBN in Figure 7(a) the operation causes an event that initiates epoch 2. According to Assumption 6, all non-instant edges are reinserted and temporal links between persistent variables, except components, are removed. In Figure 7(a), the only connection between time slices (1) and (2) are through nodes $\mathbf{c}^{(2)} = (c_{19}^{(2)}, c_{20}^{(2)})$, i.e. $\mathbf{c}^{(2)}$ d-separates the all other nodes of time slice 2 from the previous time slices. The probabilities (16) of the observations are conditioned on $\mathbf{c}^{(2)}$, and are thus independent of the previous time slices. Translating this to one single BN, we obtain B^2 in Figure 7(d).

Summarizing the manipulations on B^1 to obtain B^2 we have

- Set $B^2 = B^1$.
- Insert a non-instant edge between C_{19} and O_{25}.
- Reset the CPD of O_{24} to $p(O_{24}|pa_{B^0}(O_{24}))$.

7.2 BN Updating Algorithm

In the example in the previous section we started with a BN B^0, and manipulated this by adding and removing edges and updating CPDs as events occurred. We obtained the two BNs B^1 and B^2 that, by construction, satisfies (17). In this section we generalize the updating rules derived above, and present an algorithm *updateBN* that generalizes the manipulations to all kinds of sequences of action results. The algorithm $B^t = \textit{updateBN}(B^{t-1}, a^t, \mathbf{b}^{t-1})$ takes a BN B^{t-1}, for which (17) holds, and an action a^t as input, and delivers a BN B^t that satisfies (17). The algorithm, defined in Algorithm 1 consists of three cases depending on whether a^t is an observation, operation, or repair. We give an

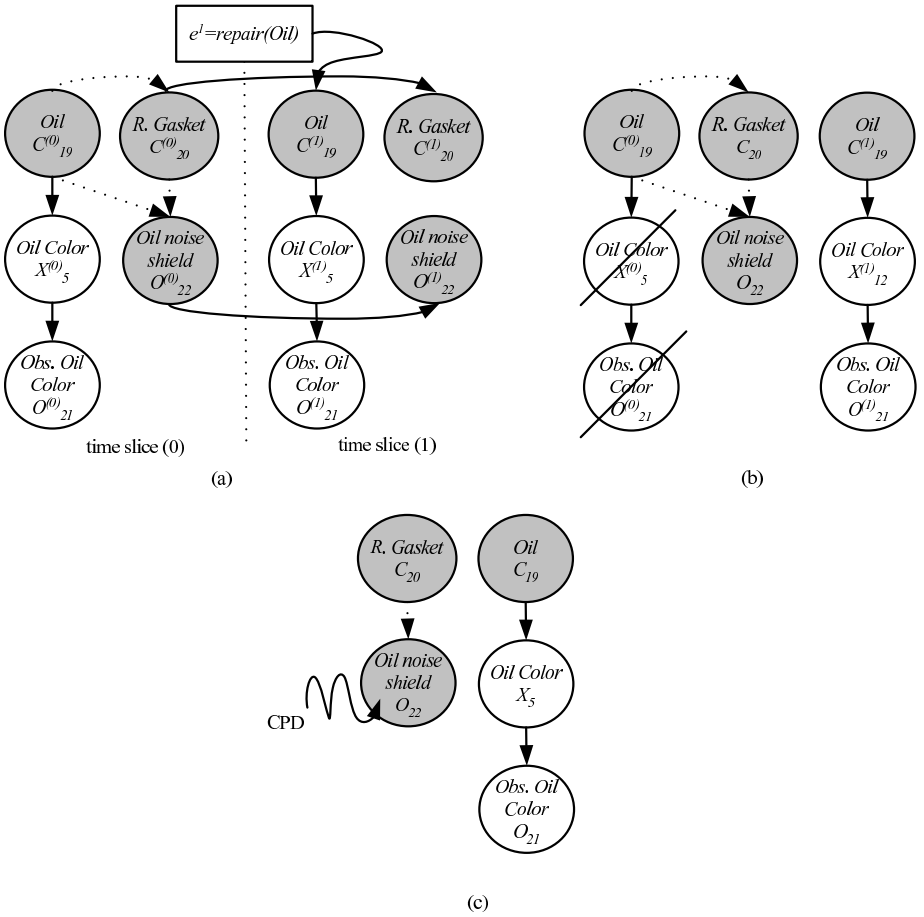


Figure 8: Merging two epochs in an nsDBN to one BN. The nsDBN is shown in (a). In (b) nodes with the same value are merged, and the child nodes of $C_{19}^{(0)}$ are crossed over since they will not contribute to the probabilities of $o_{25}^{(1)}$ and $o_{24}^{(1)}$ conditioned on $\mathbf{c}^{(0)}$. The BN in (c) is obtained by updating the CPD for $O_{25} = O_{25}^{(2)}$ with the contribution from $C_{19}^{(0)}$ and removing node $c_{19}^{(0)}$.

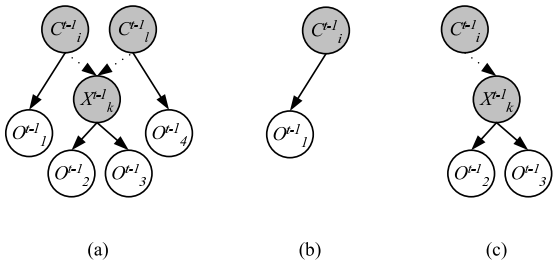


Figure 9: A troubleshooting BN in (a), the repair-influenced BN $B(C_i, O_1)$ in (b), and the repair-influenced BN $B(C_i, O_2)$ in (c).

overview of the operation and observation cases, and of the repair case, respectively, in the following two sections . We then present the complete algorithm. In this section, recall that the BNs we consider are troubleshooting BNs as described in Section 3.4.

Updating for Observation and Operation

An observation action is not an event. Thus there are no structure changes, and the algorithm *updateBN* generates a BN B^t such that $B^t = B^{t-1}$.

By Assumption 6, an operation actions basically resets the BN to the initial BN, so for an operation *updateBN* gives $B^t = B^0$.

Updating for Repairs

For repair actions, the situation is more involved. To describe the effects, we will study the effects of repair actions on subparts of the BN, called repair-influenced BNs, and defined as follows.

Definition 4 (Repair-influenced BN)**.** *A* Repair-influenced BN *in a troubleshooting BN B for component C_i and observation $O_j \in de_B(C_i)$ is denoted $B(C_i, O_j)$ and is the subpart BN consisting of the variables $\{O_j, \mathbf{R}_B(\mathbf{C}, O_j), C_i\}$ and the edges between these variables. The set $\mathbf{R}_B(\mathbf{C}, O_j)$ consists of the variables that are not d-separated from O_j by $\mathbf{C}$ in B.*

To exemplify a repair-influenced BN, Figure 9(a) shows a BN, and Figure 9(b) and (c) show the repair-influenced BNs for $B(C_i, O_1)$ and $B(C_i, O_2)$ respectively.

The repair-influenced BNs $B(C_i, O_j)$ can be classified into *structure classes*, depending on their structural properties. The elements in a structure class share structural properties, described in column two in Table 1, but the number of nodes may be different. For example "Persistent observation with an instant edge from its parent component" is one structure class. In this work we define

nine structure classes, each of them corresponding to one row in Table 1. A set of structure classes is called a *family of structure classes* and denoted $\mathcal{F}$, and in particular we let $\mathcal{F}^*$ be the family consisting of the structure classes in Table 1.

We say that a troubleshooting BN B belongs to a family $\mathcal{F}$ of structure classes if every repair-influenced BN in B belongs to a structure class in family $\mathcal{F}$. We will from now on only consider troubleshooting BNs such that the BN B^0 modeling the system when troubleshooting begins belongs to the family $\mathcal{F}^*$. This may seem technical and limiting, but since BNs belonging to $\mathcal{F}^*$ capture several kinds of component-observation relations, they are useful in many troubleshooting applications. In particular, it can be realized that the BN for the retarder, show with all instant/non-instant edges and persistent/non-persistent variables in Figure 10 belongs to $\mathcal{F}^*$.

An important property of family $\mathcal{F}^*$ is that its structure classes are constructed so that removing edges in a repair-influenced BN that belongs to a class in $\mathcal{F}^*$ will give a new repair-influenced BN that belongs to one of the nine classes in $\mathcal{F}^*$.

In Table 1 the effects of repairing C_i for the nine structure classes in $\mathcal{F}^*$ are shown. In column three of Table 1, for each structure class, a typical repair-influenced BN $B^{t-1}(C_i, O_j)$ is shown. Assume that (17) holds for this BN and let $a^t = \mathit{repair}(C_i)$. Then, as will be verified in the remainder of this section, equality (17) holds also for the corresponding BN B^t in column four of Table 1. In particular, if $B_{ns}^{(t-1:t)}$ is a two-time-slice nsDBN with initial time slice $B_{ns}^{(t-1)} = B^{t-1}$ and a second time slice $B_{ns}^{(t)}$ generated according to the assumptions in Section 5.3, the BNs B^t are such that equality

$$p(o_j^{(t)}|\mathbf{c}^{(t)}, B_{ns}^{(t-1:t)}) = p(O_j = o_j^{(t)}|\mathbf{C} = \mathbf{c}^{(t)}, B^t) \tag{21}$$

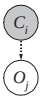
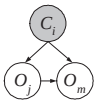
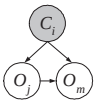
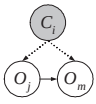
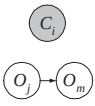
holds.

Table 1: Structure classes in family $\mathcal{F}^*$, and their updates after $a^t = \mathit{repair}(C_i)$.

No	Property	B^{t-1}	B^t	Comment
1	Component without children.			No edges to add or remove

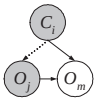
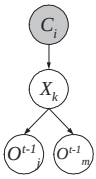
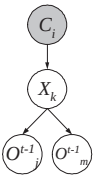
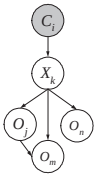
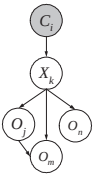
Continued on next page...

Table 1: Structure Classes of family $\mathcal{F}^*$ – continued from previous page

No	Property	B^{t-1}	B^t	Comment
2	Non-persistent observation with instant edge from parent component.			$\mathbf{C}^t$ d-separates O^t from the previous epoch.
3	Non-persistent observation with non-instant edges from parent component.			$\mathbf{C}^t$ d-separates O^t from the previous epoch.
4	Persistent observation with non-instant edges to parent component.			The CPD for O is updated to take the affect of C_i^{t-1} into account.
5	Dependent non-persistent components with instant edges to parent component. Only two observations are allowed to be directly connected.			C_i^t d-separates O_j^t and O_m^t from the previous epoch.
6	Dependent non-persistent components with non-instant edges to parent components. Only two observations are allowed to be directly connected.			C_i^t d-separates O_j^t and O_m^t from the previous epoch.

Continued on next page...

Table 1: Structure Classes of family $\mathcal{F}^*$ – continued from previous page

No	Property	B^{t-1}	B^t	Comment
7	Dependent observations, non-persistent observation with instant edge and persistent observation with non-instant edges. Only two observations are allowed to be directly connected.			The CPD for O_j is updated to take the effects of C_i^{t-1} and O_m^{t-1} into account.
8	Non-persistent internal state, instant edge from the component and to the observations. More than two child observations are allowed.			C_i^t d-separates O^t from the previous epoch.
9	Non-persistent internal state, instant edge from the component and to the observations. More than two child observations are possible, but each observation is directly connected to at most one other observation.			C_i^t d-separates O^t from the previous epoch.

Structure Class 1. The manipulations on B^{t-1} to obtain B^t are trivial for structure class 1, since there is one single component without children. In this case there are no edges to add or remove.

Structure Classes 2, 3, 5, 6, 8, and 9. For structure classes 2, 3, 5, 6, 8, and 9, the common factor is that C_i has non-persistent descendants only. This means that, as in the computation of the probability of $O_{24}^{(1)}$ in (18), the observations made after the repair are independent of the previous actions since

c_i^t after the repair action is given. To obtain B^t from B^{t-1}, we set $B^t := B^{t-1}$. We then remove all non-instant edges in B^t.

Structure Classes 4 and 7. Structure classes 4 and 7 share the property that the children of C_i are observable symptoms, and that at least one of them is persistent. Similarly to the computations for $O_{25}^{(1)}$ in (19) and (20), the BN B^t is obtained from B^{t-1} by removing all non-instant edges, and updating the CPD for the persistent observable symptom variables O_j to take information from the previous time slice into account. To determine the updated CPD, note that (17) holds for $B_{ns}(\mathbf{e}^{1:t-1})$ and B^{t-1}. Note also that $pa_{B^t}(O_j) = \emptyset$. We search an updating of the CPD for O_j after the repair such that $p(o_j^{(t)}|pa_{B^t}(o_j), B^t) = p(o_j^{(t)}|B^t) = p(O_j^{(t)} = o_j|\mathbf{a}^{1:t}, B_{ns}(\mathbf{e}^{1:t}))$. Consider the last probability in the equality. Marginalizing over $C_i^{(t-1)}$ gives

$$\begin{aligned} &p(o_j^{(t)}|\mathbf{a}^{1:t}, B_{ns}(\mathbf{e}^{1:t})) = \\ &= \sum_{c_i^{(t-1)}} p(o_j^{(t)}|c_i^{(t-1)}, \mathbf{a}^{1:t}, B_{ns}(\mathbf{e}^{1:t}))p(c_i^{(t-1)}|\mathbf{a}^{1:t}, B_{ns}(\mathbf{e}^{1:t})) = \\ &= \sum_{c_i^{(t-1)}} \underbrace{p(o_j^{(t)}|c_i^{(t-1)}, \mathbf{a}^{1:t-1}, B_{ns}(\mathbf{e}^{1:t-1}))}_{\text{(a)}} \underbrace{p(c_i^{(t-1)}|\mathbf{a}^{1:t-1}, B_{ns}(\mathbf{e}^{1:t-1}))}_{\text{(b)}}. \end{aligned} \quad (22)$$

To obtain probability (a) in the sum (22), we have used that $O^{(t)}$ is independent of the future repair of C_i given $C_i^{(t-1)}$. For probability (b) in the sum (22) we have used that a^t is an external intervention performed *after* $C_i^{(t-1)}$, which means that $C_i^{(t-1)}$ is independent of a^t. The probability (b) can be recognized as the previous belief state $\mathbf{b}^{t-1}$, and is known. For the first probability in the sum we have, by using (16) and then (17), that

$$\begin{aligned} &p(o_j^{(t)}|c_i^{(t-1)}, \mathbf{a}^{1:t-1}) = p(o_j^{(t-1)}|c_i^{(t-1)}, \mathbf{a}^{1:t-1}) = \\ &= p(o_j^{(t-1)}|c_i^{(t-1)}, \mathbf{v}^{1:t-1}, B_{ns}(\mathbf{e}^{1:1-t})) = \quad (23) \\ &p(O_j = o_j^{(t-1)}|C_i = c_i^{(t-1)}, \mathbf{v}^{1:t-1}, B^{t-1}), \quad (24) \end{aligned}$$

where we in the last equality have used (17). To summarize, from (22) and (23) we have that the CPD for O_j is computed using its CPD in B^{t-1} and the previous belief state $\mathbf{b}^{t-1}$.

Updating Algorithm

Pseudo-code for the algorithm *updateBN* is given in Algorithm 1. For a BN B^0 that belongs to $\mathcal{F}^*$, and given a sequence $\mathbf{a}^{1:t}$ of action results, *updateBN* generates a sequence $B^1, \ldots, B^t$ of BNs that each satisfies (17). The algorithm consists of three cases (if-statements), one for observation actions, one for operation actions, and one repair actions. Within the if-statement for repair actions,

the set *ConsideredObs* is constructed to avoid that the same repair-influenced BN is considered several times. The following theorem guarantees the properties of the updating algorithm *updateBN* defined by Algorithm 1.

Theorem 1 (Algorithm *updateBN*)**.** *Consider a troubleshooting session described by an nsDBN B_{ns} with initial BN B_{ns}^0 belonging to $\mathcal{F}^*$ and a sequence $\mathbf{a}^{1:t}$ of action results such that there is at lest one operation action between two repairs actions. Let $B^0 = B_{ns}^0$ and let $B^1, \ldots, B^t$ be a sequence of BNs such that $B^k = updateBN(B^{k-1}, a^k), k = 1, \ldots t$, where updateBN is defined by Algorithm 1. Then,* (17) *holds for each B^k, $k = 0, \ldots t$.*

The theorem is proved in the Appendix. The proof includes many technical details, and is not necessary for the application of the algorithm.

Algorithm 1 $B = updateBN(B^-, a)$

```
B := B⁻
if a = observe(O = o) then
  // Nothing to do
else if a^t = operate then
  B := B^0
else if a = repair(C) then
  ConsideredObs := ∅
  for all O ∈ de(C) do
    if O ∉ ConsideredObs then
      Ω := {O' : O' ∈ B(C, O)}
      ConsideredObs := ConsideredObs ∪ Ω
      Update B(C, O) according to Table 1
    end if
  end for
end if
```

8 Modeling Application

The troubleshooting system consisting of a planner and a diagnoser as described in Sections 4-7 is implemented and applied to the problem of troubleshooting a heavy truck with a faulty retarder. A BN B^0 modeling the retarder at arrival to the workshop is shown in Figure 10. This model is built from expert knowledge, and by applying the modeling principles developed in Section 5. The retarder BN belongs to $\mathcal{F}^*$, so the algorithm *updateBN* is applicable. In Figure 10, instant edges are solid, non-instant edges are dotted, persistent nodes are gray, and non-persistent nodes are white.

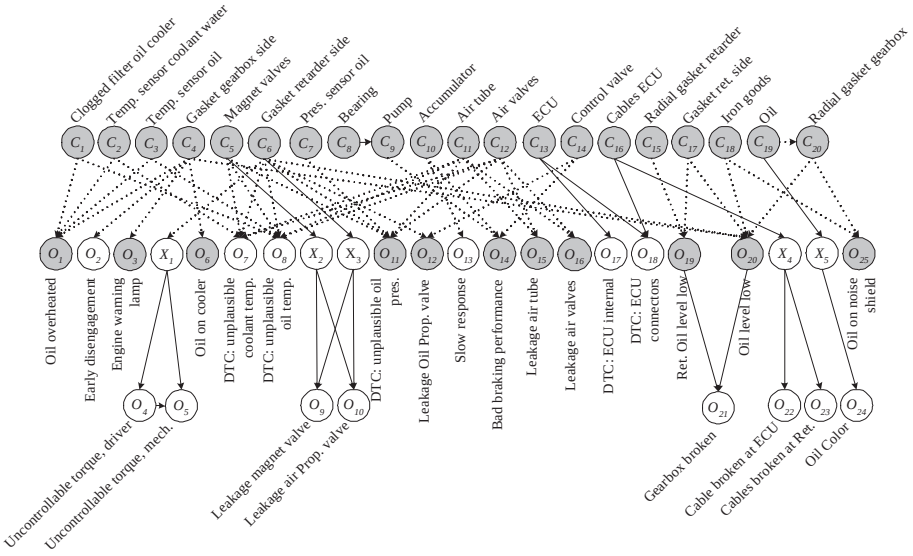


Figure 10: A Bayesian network modeling the retarder.

In the implementation, the size of the belief state with which the planner is initialized, is limited such that only the 21 most probable diagnoses $\mathbf{c}$ of component statuses are kept. Also, the diagnoser is set to disregard diagnoses where four or more components are faulty. This is done to keep the size of the belief state manageable, and it is reasonable because the probability for several simultaneous faults in the retarder is typically very small compared to having fewer faults. This method of keeping down the size of the belief state works for our model of the retarder, but it is not feasible for larger systems. In those cases methods as the one presented in [Lerner et al., 2000] can be used, where the diagnoser collapses similar diagnoses into one.

To investigate the relevance of accurate probability computations by the diagnoser, we introduced noise in the parameters in the BN. Noise is added using the log-odds normal distribution as described in [Kipersztok and Wang, 2001]. Every parameter θ in the CPDs inB^0 receives a new value θ' which is

$$\theta' = \frac{1}{1 + (\theta^{-1} - 1) \cdot 10^{-\omega_\sigma}}, \tag{25}$$

where ω_σ is a random number drawn from a normal distribution with standard deviation σ. The troubleshooter has only access to this distorted model, while an undistorted model of the retarder is used to represent the physical system. In each test case there is a predefined fault. When actions are performed, the results are drawn randomly in accordance with the undistorted model and the predefined fault. The troubleshooting process is simulated until the fault

is repaired and the total cost is measured. To avoid long waiting times, the planner is aborted after 60 seconds of deliberation. The standard deviation σ is varied from 0 to 1, and for each level of σ, 25 test cases are run. Figure 11 shows the average discrepancy in the cost for troubleshooting using the noisy BN and compared to using the nominal BN. Small errors in the parameters does not effect the result significantly, but for noise with standard deviation above 0.25 the error increases fast. In practice, the result in Figure 11 means that, since small parameters errors have an (almost) insignificant impact on the ECR computed, the parameters could be chosen roughly.

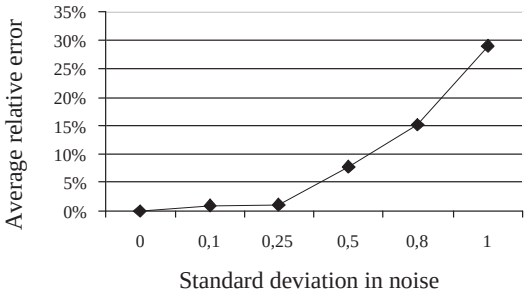


Figure 11: Average error in the expected cost of repair when the parameters in the BN are distorted.

9 Conclusion and Future Work

9.1 Conclusion

Inspired by a case study of the retarder, an auxiliary heavy truck breaking system, a decision theoretic troubleshooting system has been developed. Focus has been on issues important in real world applications: the need for disassembling the system during troubleshooting, the problem of verifying that the system is fault free during the troubleshooting, and the fact that computations for suggestion of new actions should be performed while the mechanic is waiting. These issues have two main consequences: probabilities must be computed in a system that is subject to external interventions, and the computations should be fast.

The troubleshooting system developed is based on a decision-theoretic approach. It consists of a planner that suggests the next troubleshooting action to the mechanic, and a diagnoser that supports the planner with probabilities for faults. In the planner, an any-time AO^* algorithm with heuristics has been used. In the diagnoser, probabilities are efficiently computed by an algorithm,

based on a static BN, and consisting of two parts: belief state updating and model updating.

Driven by the application study of the retarder, we have studied practical issues of modeling for troubleshooting in detail, and provided guidelines for building the BN to be used in the diagnoser. In particular, there are two different types of dependencies that are used in troubleshooting: instant and non-instant dependencies. To handle this fact, in combination with the need for handling the external interventions caused by repairs and operations and the need for time efficient computations, a new algorithm *updateBN* has been developed. The algorithm *updateBN* reduces the external interventions to simple manipulations on a static BN. The manipulated BN does not model the troubleshooting system in a general way, but has been proved to compute the probabilities needed in the diagnoser correctly.

Finally, we have applied the troubleshooting system to the retarder. The results confirm the suggested modeling approach and that the decision theoretic troubleshooting approach used is suitable in real-world applications.

9.2 Future Work

There are several interesting open questions for future work regarding the BN updating algorithm and the diagnoser. Most important, and our next step, is to perform a thorough validation of the efficiency of the algorithm, and compare the computation time needed when using *updateBN* with the time needed when applying event-driven nsDBN. Furthermore, intend to extend the proof of the algorithm to include sequences of action results, without the assumption that there is a an operation action between two events. We also intend to extend the family $\mathcal{F}^*$ of structure classes, and in particular to add classes containing persistent internal variables. We believe that the algorithm have capacity of handling these situations without major changes, but they should be theoretically verified.

Considering the complete troubleshooting system, one interesting question for future work is to investigate the distribution of computation time between the diagnoser and the planner. Since the mechanic is busy waiting for the next action to be suggested by the troubleshooting system, the computation time for each suggestion can be considered as limited. In the diagnoser, computation time is used to compute accurate probabilities. In the planner computation time is used in the search for a plan that is as close to the optimal as possible. Thus, there is a trade-off between accurate probabilities and optimal planning.

Moreover, one challenge is the dimension of the belief state, which increases exponentially with the number of components. Therefore methods for focusing on the most probable diagnoses in the diagnoser, without risking to loose diagnoses with small probabilities in the first time steps, are interesting future work.

However, as a first step towards troubleshooting with interventions and both instant and non-instant dependencies, the results presented in this work are promising, and show that computer aided troubleshooting can be applied to complex mechatronic systems such as the retarder. We look forward to extend our algorithm to troubleshoot even larger systems such as complete vehicles.

Appendix

To prove Theorem 1 we begin with Lemma 3, where we consider the special case where there are observation actions only. In Lemma 4 we extend the results to include both observations and operations. We then consider the updates generated by a single repair action in Lemma 5, and conclude by proving Theorem (1) for a general sequence of action results.

Lemma 3 (Observation Update)**.** *Consider a troubleshooting session described by an nsDBN B_{ns} with initial BN B_{ns}^0 and a sequence $\mathbf{a}^{1:t}$ of actions results from observations. Let $B^0 = B_{ns}^0$ and let $B^1, \ldots, B^t$ be a sequence of BNs such that $B^k = updateBN(B^{k-1}, a^k), k = 1, \ldots t$, where updateBN is defined by Algorithm 1. Then,* (17) *holds for each B^k, $k = 0, \ldots t$.*

Proof of Lemma 3. Observations only are considered and observations do not cause events. Thus, $B_{ns}(\mathbf{e}^{1:1}) = B_{ns}(\langle 0 \rangle) = B_{ns}^0 = B^0$. The variables $o_j^{(1)}$ and $\mathbf{c}^{(1)}$ in B_{ns} correspond to o_j and $\mathbf{c}$ in B^0. This gives:

$$p(o_j^{(1)} | \mathbf{c}^{(1)}, B_{ns}(\mathbf{e}^{1:1})) = p(O_j = o_j^{(1)} | \mathbf{C} = \mathbf{c}^{(1)}, B^0), \tag{26}$$

which verifies equality (17) for B^0. Again, since observations do not cause events, we have $B_{ns}(\mathbf{e}^{1:k}) = B_{ns}(\langle 0, \ldots, 0 \rangle) = B_{ns}(\langle 0 \rangle) = B_{ns}^0$, $k = 1, \ldots t$. For an observation a^k, *updateBN* defined by Algorithm 1 generates a BN $B^k = B^{k-1} = \ldots = B^0$, $k = 1, \ldots t$. Thus, (26), and thereby (17), holds for each B^k, $k = 1, \ldots t$. □

Lemma 4 (Operation Update)**.** *Consider a troubleshooting session described by an nsDBN B_{ns} with initial BN B_{ns}^0 and a sequence of $\mathbf{a}^{1:t}$ of action results from observations and operations. Let $B^0 = B_{ns}^0$ and let $B^1, \ldots, B^t$ be a sequence of BNs such that $B^k = updateBN(B^{k-1}, a^k), k = 1, \ldots t-1$, where updateBN is defined by Algorithm 1. Then,* (17) *holds for each B^k, $k = 1, \ldots t$.*

Proof of Lemma 4. Let a^{t_p} be the first operation action in $\mathbf{a}^{1:t}$ and consider the first subsequence $\mathbf{a}^{1:t_p}$. We first verify (17) for an observation a^{t_p+1}, i.e. for $O_j^{(t_p+1)} = O_j^{(t_p)}$ made after the first operation action, but before next operation. By Lemma 3 B^{t_p-1} satisfies (17). Let $B_{ns}^{(t_p)}$ denote the last time slice of $B_{ns}(\mathbf{e}^{1:t_p})$. Since $B_{ns}^{(t_p)}$ is caused by an operation, it is, according to Assumption 6 a copy of the initial BN B_{ns}^0 that is connected to the previous time slice

only through edges between the components. Thus, the observable symptom variables in $B_{ns}^{(t_p)}$ are d-separated from the previous time slices by $\mathbf{C}^{(t_p)}$. For the static BN B^{t_p} generated by *updateBN* defined by Algorithm 1 we have that $B^{t_p} = B_{ns}^0$. This gives:

$$p(o_j^{(t_p)}|\mathbf{c}^{(t_p)}, \mathbf{v}^{1:t_p}, B_{ns}(\mathbf{e}^{1:t_p})) = p(o_j^{(t_p)}|\mathbf{c}^{(t_p)}, B_{ns}^{(t_p)}) =$$
$$= p(o_j^{(t_p)}|\mathbf{c}^{(t_p)}, B_{ns}^0) = p(O_j = o_j^{(t_p)}|\mathbf{C} = \mathbf{c}^{(t_p)}, B^{t_p}), \quad (27)$$

where we, in the last equality, have used that the variables $O_j^{(t_p)}$ and $\mathbf{c}^{(t_p)}$ in B_{ns} correspond to O_j and $\mathbf{C}$ in B^{t_p}. By equation (27) we have that (17) holds for observations made between the first and the second operation actions.

From the first equality in (27), we have that computations are independent on the previous actions, before the repair at t_p. Thus, (27) holds even for $O_j^{(t)}$, which can have be preceded by a general sequence of observations and operations. □

Lemma 5 (Single Repair Update)**.** *Consider a troubleshooting session described by an nsDBN B_{ns} with initial BN B_{ns}^0 belonging to $\mathcal{F}^*$ and a sequence of $\mathbf{a}^{1:t} = \langle \mathbf{a}^{1:t-1}, a^t \rangle$ of action results where $\mathbf{a}^{1:t-1}$ consists of observation and operation actions only and $a^t = repair(C_i)$. Let $B^0 = B_{ns}^0$ and let $B^1, \ldots, B^t$ be a sequence of BNs such that $B^k = updateBN(B^{k-1}, a^k), k = 1, \ldots t-1$, where updateBN is defined by Algorithm 1. Then,* (17) *holds for each B^k, $k = 0, \ldots t$.*

To prove Lemma 5 we need the following lemma, that concerns the local properties of repair actions. It says that in a troubleshooting BN, observations that are not descendants to the repaired component are not affected by the repair.

Lemma 6 (Locality of *repair*)**.** *Let $a^t = repair(C_i)$, let $\{O_j^{(t-1)}, O_j^{(t)}\}$ be a pair of variables representing the same observable symptom before and after the repair a^t and such that $O_j^{(0)} \notin de_{B_{ns}(\langle 0 \rangle)}(C_i^{(0)})$. Then it holds that*

$$p(O_j^{(t)} = o|\mathbf{c}^{(t)}, \mathbf{v}^{1:t}, B_{ns}(\mathbf{e}^{1:t})) = p(O_j^{(t-1)} = o|\mathbf{c}^{(t-1)}, \mathbf{v}^{1:t-1}, B_{ns}(\mathbf{e}^{1:t-1})). \quad (28)$$

Proof of Lemma 6. By Assumption 5, $a^t = repair(C_i)$ affects C_i only. Thus,

$$\mathbf{C}_{\bar{i}}^{(t)} = \mathbf{C}_{\bar{i}}^{(t-1)}. \quad (29)$$

In $B_{ns}(\mathbf{e}^{1:t})$, the last time slice $B_{ns}^{(t)}$, caused by $a^t = repair(C_i)$, is the same as for the empty event but without the edge from $C_i^{(t-1)}$ to $C_i^{(t)}$. This means that the repair introduces no edges between variables in $B_{ns}^{(t)}$ that are not present in $B_{ns}^{(0)} = B_{ns}(\langle 0 \rangle)$. Thus, $O_j^{(0)} \notin de_{B_{ns}(\langle 0 \rangle)}(C_i^{(0)})$ gives that $O_j^{(t)} \notin de(C_i^{(t)})$

and $O_j^{(t-1)} \notin de(C_i^{(t-1)})$. Since $B^0 = B^{(0)}$ belongs to $\mathcal{F}^*$, we there are at most two directly dependent observable symptoms, required that they also have the same ancestors. Thus, there is not path between a variable in $\{O_j^{(t-1)}, O_j^{(t)}\}$ to a variable in $\{C_i^{(t-1)}, C_i^{(t)}\}$ that not passes through $\mathbf{C}_{\bar{i}}^{(t-1)} \cup \mathbf{C}_{\bar{i}}^{(t)}$. This makes $\{O_j^{(t-1)}, O_j^{(t)}\}$ d-separated from $\{C_i^{(t-1)}, C_i^{(t)}\}$ by $\mathbf{C}_{\bar{i}}^{(t-1)} \cup \mathbf{C}_{\bar{i}}^{(t)}$ and we have that

$$p(O_j^{(t)} = o|\mathbf{c}^{(t)}, \mathbf{v}^{1:t}, B_{ns}(\mathbf{a}^{1:t})) = p(O_j^{(t)} = o|\mathbf{c}_{\bar{i}}^{(t)}, \mathbf{v}^{1:t}, B_{ns}(\mathbf{a}^{1:t})) =$$
$$= p(O_j^{(t-1)} = o|\mathbf{c}_{\bar{i}}^{(t-1)}, \mathbf{v}^{1:t}, B_{ns}(\mathbf{a}^{1:t})) = p(O_j^{(t-1)} = o|\mathbf{c}^{(t-1)}, \mathbf{v}^{1:t-1}, B_{ns}(\mathbf{a}^{1:t-1})),$$

where we in the second equality have used that, by the construction of the new time slice, observation $O_j^{(t-1)}$ has the same relations to $\mathbf{c}_{\bar{i}}^{(t-1)}$ as $O_j^{(t)}$ has to $\mathbf{c}_{\bar{i}}^{(t)}$. This proves Lemma 6. □

We are now ready to prove Lemma 5.

Proof of Lemma 5. To prove that (17) holds for an observation $O_j^{(t)}$, and a sequence of actions $\mathbf{a}^{1:t}$ where $\mathbf{a}^{1:t-1}$ are observations and operations, and $a^t = \mathit{repair}(C_i)$, we consider two cases: Case 1, where $O_j^{(t)}$ is such that $O_j^{(0)} \notin de_{Bns}(C_i^{(0)})$; and Case 2, where $O_j^{(t)}$ is such that $O_j^{(0)} \in de_{Bns}(C_i^{(0)})$.

Case 1. For Case 1, (28) holds according to Lemma 6, i.e. the distribution for $O_j^{(t)}$ is equal to the distribution for $O_j^{(t-1)}$ given $\mathbf{v}^{1:t} = \langle \mathbf{v}^{1:t}, 0 \rangle$ and $B_{ns}(\mathbf{e}^{1:t})$ and $B_{ns}(\mathbf{e}^{1:t-1})$ respectively. Next, study the updating of B^{t-1} to B^t by *updateBN*. Since $O_j^{(0)} \notin de_{Bns}(C_i^{(0)})$ we have that $O_j \notin de_{B^0}(C_i)$, and, since B^0 contains all possible edges, that $O_j \notin de_{B^k}(C_i)$, $k = 1, \ldots, t$. Recall that the structure of the BNs B^k is such that the direction of all paths is from components towards observations, possibly passing internal variables, and that dependent observations have the same ancestors. This means that there is no repair-influenced BN $B^k(C_i, O_l)$, $k = 1, \ldots, t$ such that $O_j \in B^k(C_i, O_l)$. In algorithm *updateBN*, when $a^t = \mathit{repair}(C_i)$, there are updating manipulations only on the repair-influenced BNs $B^k(C_i, O_l)$ for C_i and $O_l \in de_{B^k}(C_i)$. Thus,

$$p(o_j|\mathbf{c}, \mathbf{v}^{1:t}, B^t) = p(o_j|\mathbf{c}, \mathbf{v}^{1:t-1}, B^{t-1}) \tag{30}$$

By Lemmas 3 and 4 we have that (17) holds for $O_j^{(t-1)}$. Using this, together with (28) and (30) we have that

$$p(o_j^{(t)}|\mathbf{c}^{(t)}, \mathbf{v}^{1:t}, B_{ns}(\mathbf{e}^{1:t})) = p(o_j^{(t-1)}|\mathbf{c}^{(t-1)}, \mathbf{v}^{1:t-1}, B_{ns}(\mathbf{e}^{1:t-1})) =$$
$$= p(O_j = o_j^{(t-1)}|\mathbf{c}, \mathbf{v}^{1:t-1}, B^{t-1}) = p(O_j = o_j^{(t-1)}|\mathbf{c}, \mathbf{v}^{1:t}, B^t), \tag{31}$$

which proves that (17) holds for $O_j^{(t)}$, and that Lemma 5 holds for Case 1.

Case 2. In Case 2 we consider variables $O_j^{(0)} \in de_{Bns}(C_i^{(0)})$. Recall that an observation action $a^{t_{ob}}$ does not affect the structure of the BN, i.e. we have that $B_{ns}^{(t_{ob})} = B_{ns}^{(t_{ob}-1)}$ and $B^{t_{ob}} = B^{t_{ob}-1}$. An operation action $a^{t_{op}}$ results

in a time slice that is equal to the initial time slice, i.e. $B_{ns}^{(t_{op})} = B_{ns}^{(0)}$ and $B^{t_{op}} = B^0$. Thus, after the action sequence $\mathbf{a}^{1:t-1}$, that consists of observations and operations only, we have $B_{ns}^{(t-1)} = B_{ns}^{(0)}$ and $B^{t-1} = B^0$. Thus, the action sequence $\mathbf{a}^{1:t}$ generates an nsDBN with the two time slices $B_{ns}^{(t-1)}$ and $B_{ns}^{(t)}$, and where the structure in the first time slice, $B_{ns}^{(t-1)}$, is equal to the initial BN. This means that we have that $O_j^{(t-1)} \in de_{Bns}(C_i^{(t-1)})$.

The repair-influenced BN $B_{ns}^{(t-1)}(C_i^{(t-1)}, O_j^{(t-1)})$ in time slice $B_{ns}^{(t-1)}$ is, by Definition 4, constructed so that, given $\mathbf{C}^{(t-1)}$, the variable $O_j^{(t-1)}$ in $B_{ns}^{(t-1)}$ is independent of the variables in[3] $B_{ns}^{(t-1)} \setminus B_{ns}^{(t-1)}(C_i^{(t-1)}, O_j^{(t-1)})$.

Let $\tilde{B}_{var}$ be a set of variables such that for each variable

$$Y^{(t-1)} \in B_{ns}^{(t-1)}(C_i^{(t-1)}, O_j^{(t-1)}),$$

then, for its corresponding variable in the next time slice we have $Y^{(t)} \in \tilde{B}_{var}$. Let $\tilde{B}$ be the BN consisting of the variables $\tilde{B}_{var}$ and their edges in $B_{ns}^{(t)}$.

Since repair actions remove edges, no edges can be added in $B_{ns}^{(t)}$ compared to $B_{ns}^{(t-1)}$. Furthermore, since $B_{ns}^{(t-1)}(C_i^{(t-1)}, O_j^{(t-1)})$ is constructed so that, given $\mathbf{C}^{(t-1)}$, the variable $O_j^{(t-1)}$ in $B_{ns}^{(t-1)}$ is independent of the variables $B_{ns}^{(t-1)} \setminus B_{ns}^{(t-1)}(C_i^{(t-1)}, O_j^{(t-1)})$, then it also holds that $O_j^{(t)}$ in $B_{ns}^{(t)}$ is independent of the variables $B_{ns}^{(t)} \setminus \tilde{B}_{var}$ given $\mathbf{C}^{(t)}$. In words, this means that when $\mathbf{C}^{(t-1)}$ and $\mathbf{C}^{(t)}$ are given, $O_j^{(t-1)}$ and $O_j^{(t)}$ are, within their corresponding time slice, dependent only on variables in $B_{ns}^{(t-1)}(C_i^{(t-1)}, O_j^{(t-1)})$ and $\tilde{B}$, respectively.

Now, we will study how information flows between the two time slices. Recall that in the troubleshooting BN, the edges between time slices, the so called temporal edges, only connect a variable with its copy in the adjacent time slice. This means for example that there may be an edge between $O_l^{(t-1)}$ and $O_l^{(t)}$, but not between $O_l^{(t-1)}$ and $O_m^{(t)}$ if $l \neq m$. Thus, the variables in $B_{ns}^{(t-1)}(C_i^{(t-1)}, O_j^{(t-1)})$ have temporal edges only to variables in $\tilde{B}$.

The reasoning above means that, given $\mathbf{C}^{(t)}$ and $\mathbf{C}^{(t-1)}$, the variable $O_j^{(t)}$ is, in $B_{ns}(\mathbf{e}^{1:t})$, which consists only of the two time slices $B_{ns}^{(t-1)}$ and $B_{ns}^{(t)}$ and the temporal edges between them, dependent only on nodes in the subpart of $B_{ns}(\mathbf{e}^{1:t})$ consisting of $B_{ns}^{(t-1)}(C_i^{(t-1)}, O_j^{(t-1)})$, $\tilde{B}$, and the temporal edges between them. Denote this subpart BN $B_{ns}^{(t-1)}(C_i^{(t-1)}, O_j^{(t-1)}) \rightarrow \tilde{B}$. Since repairs are local and only affects the repaired component we have that $C_{\bar{i}}^{(t-1)} = C_{\bar{i}}^{(t)}$. Then, by using marginalization over $C_i^{(t-1)}$, we can write the left hand

[3]Whenever we us set relations and set operations on BNs, the BNs are considered as sets of variables.

side of (17) as

$$p(o_j^{(t)}|\mathbf{c}^{(t)},\mathbf{v}^{1:t},B_{ns}(\mathbf{e}^{1:t})) = \\ = \sum_{c_i^{(t-1)}} \underbrace{p(o_j^{(t)}|\mathbf{c}^{(t)},c_i^{(t-1)},\mathbf{v}^{1:t},B_{ns}(\mathbf{e}^{1:t}))}_{\text{(a)}} \underbrace{p(c_i^{(t-1)}|\mathbf{c}^{(t)},\mathbf{v}^{1:t},B_{ns}(\mathbf{e}^{1:t}))}_{\text{(b)}}. \quad (32)$$

For the probability (a) in (32) we have, by the reasoning above, that

$$p(o_j^{(t)}|\mathbf{c}^{(t)},c_i^{(t-1)},\mathbf{v}^{1:t},B_{ns}(\mathbf{e}^{1:t})) = p(o_j^{(t)}|\mathbf{c}^{(t)},\mathbf{c}^{(t-1)},\mathbf{v}^{1:t},B_{ns}(\mathbf{e}^{1:t})) = \\ = p(o_j^{(t)}|\mathbf{c}^{(t)},\mathbf{c}^{(t-1)},\mathbf{v}^{1:t},B_{ns}^{(t-1)}(C_i^{(t-1)},O_j^{(t-1)}) \to \tilde{B}) = \\ = p(o_j^{(t)}|\mathbf{c}^{(t)},c_i^{(t-1)},\mathbf{v}^{1:t},B_{ns}^{(t-1)}(C_i^{(t-1)},O_j^{(t-1)}) \to \tilde{B}). \quad (33)$$

The probability (b) in (32) can be computed from the previous belief state as follows:

$$p(c_i^{(t-1)}|\mathbf{c}^{(t)},\mathbf{v}^{1:t},B_{ns}(\mathbf{e}^{1:t})) = p(c_i^{(t-1)}|\mathbf{c}_{\bar{i}}^{(t-1)},\mathbf{v}^{1:t-1},B_{ns}(\mathbf{e}^{1:t-1})) = \\ = p(c_i^{(t-1)}|\mathbf{c}_{\bar{i}}^{(t-1)},\mathbf{a}^{1:t-1}), \quad (34)$$

where we in the first equality have used that $\mathbf{c}_{\bar{i}}^{(t-1)} = \mathbf{c}_{\bar{i}}^{(t)}$, that $v^t = 0$ since a^t is a repair, and that $c_i^{(t-1)}$ is independent of the future repair in the troubleshooting BNs considered here. Note that in (34) we can add the subnetwork $B_{ns}^{(t-1)}(C_i^{(t-1)},O_j^{(t-1)}) \to \tilde{B}$ without changing the probabilities. Using this expansion, and inserting (33) and (34) into (32) gives:

$$p(o_j^{(t)}|\mathbf{c}^{(t)},\mathbf{v}^{1:t},B_{ns}(\mathbf{e}^{1:t})) = \\ \sum_{c_i^{(t-1)}} p(o_j^{(t)}|\mathbf{c}^{(t)},c_i^{(t-1)},\mathbf{v}^{1:t},B_{ns}^{(t-1)}(C_i^{(t-1)},O_j^{(t-1)})\to\tilde{B})p(c_i^{(t-1)}|\mathbf{c}_{\bar{i}}^{(t-1)},\mathbf{a}^{1:t-1}) = \\ = \sum_{c_i^{(t-1)}} p(o_j^{(t)}|\mathbf{a}^{1:t-1},\mathbf{c}^{(t)},c_i^{(t-1)},\mathbf{v}^{1:t},B_{ns}^{(t-1)}(C_i^{(t-1)},O_j^{(t-1)})\to\tilde{B}) \times \ldots \\ p(c_i^{(t-1)}|\mathbf{c}_{\bar{i}}^{(t-1)},\mathbf{a}^{1:t-1},\mathbf{v}^{1:t},B_{ns}^{(t-1)}(C_i^{(t-1)},O_j^{(t-1)})\to\tilde{B}) \\ = \sum_{c_i^{(t-1)}} p(o_j^{(t)}|\mathbf{a}^{1:t-1},\mathbf{c}^{(t)},c_i^{(t-1)},\mathbf{v}^{1:t},B_{ns}^{(t-1)}(C_i^{(t-1)},O_j^{(t-1)})\to\tilde{B}) \times \ldots \\ p(c_i^{(t-1)}|\mathbf{c}^{(t)},\mathbf{a}^{1:t-1},\mathbf{v}^{1:t},B_{ns}^{(t-1)}(C_i^{(t-1)},O_j^{(t-1)})\to\tilde{B}) \\ = p(o_j^{(t)}|\mathbf{c}^{(t)},\mathbf{v}^{1:t},B_{ns}^{(t-1)}(C_i^{(t-1)},O_j^{(t-1)})\to\tilde{B}), \quad (35)$$

where we, in the second equality, have used that in the second factor we can condition on $\mathbf{v}^{1:t}$, since it is redundant to $\mathbf{a}^{1:t-1}$ and since $\mathbf{v}^t = 0$. In the first factor we can condition on $\mathbf{a}^{1:t-1}$ since all relevant information in that probability is contained in the two-time slice BN and the evidence given. The

same reasoning applies when removing $\mathbf{a}^{1:t}$ in the last equality. In the second factor of the third equality of (35), we once again used that $c_i^{(t-1)}$ is independent on its future value after the repair in the BNs we consider and conditioned on $\mathbf{c}_{\tilde{i}}^{(t-1)} = \mathbf{c}_{\tilde{i}}^{(t)}$, together with this last equality, to reinsert $\mathbf{c}^{(t)}$.

Since $B_{ns}^{(t-1)} = B_{ns}^{(0)}$ and $B^{t-1} = B^0$, and it is given that $B_{ns}^{(0)} = B_{ns}^0 = B^0$, the variables in $B_{ns}^{(t-1)}(C_i^{(t-1)}, O_j^{(t-1)})$ correspond to the variables in the repair-influenced BN $B^{t-1}(C_i, O_j)$, and $\tilde{B}$ contains the corresponding variables in B^t. Since B^{t-1} belongs to $\mathcal{F}^*$, the repair-influenced $B^{t-1}(C_i, O_j)$ belongs to one of the structure classes in family $\mathcal{F}^*$. The updating rules for the repair-influenced BNs in $\mathcal{F}^*$ are given in Table 1 and derived in Section 7.2, and, noting that $\mathbf{v}^{1:t}$ is evidence that does not affect the structure of the BNs, we can apply (21) to obtain

$$p(o_j^{(t)}|\mathbf{c}^{(t)}, \mathbf{v}^{1:t}, B_{ns}^{(t-1)}(C_i^{(t-1)}, O_j^{(t-1)}) \rightarrow \tilde{B}) = p(O_j = o_j^{(t)}|\mathbf{C} = \mathbf{c}^{(t)}, \mathbf{v}^{1:t}, \tilde{B}^t), \tag{36}$$

where $\tilde{B}^t$ a BN consisting of variables corresponding to those in $\tilde{B}$. By the construction of $\tilde{B}^t$, we have that the repair-influenced BN $B^t(C_i, O_j)$ is a subpart of $\tilde{B}^t$. Thus, O_j in B^t is independent of all variables $B^t \setminus \tilde{B}^t$, and we have that $p(O_j = o_j^{(t)}|\mathbf{C} = \mathbf{c}^{(t)}, \mathbf{v}^{1:t}, \tilde{B}^t) = p(O_j = o_j^{(t)}|\mathbf{C} = \mathbf{c}^{(t)}, \mathbf{v}^{1:t}, B^t)$, which proves Lemma 5 for Case 2. □

Finally, we turn to the proof of Theorem 1, where we show that the algorithm *updateBN*, for a general sequence of action results $\mathbf{a}^{1:t}$ gives a sequence $B^1, \ldots, B^t$ of BNs that each satisfies (17).

Proof of Theorem 1. From Lemmas 3, 4, and 5 we know that (17) holds for a sequence $B^1, \ldots, B^{t_p}$ of BNs generated by *updateBN* and an action sequence $\mathbf{a}^{1:t_p}$, where $\mathbf{a}^{1:t_p-1}$ consists of observations and operations only, and $a^{t_p} = \mathit{repair}(C_i)$.

Assume that (17) holds for a sequence of BNs $B^1, \ldots, B^{t_q}$, where $a^{t_q} = \mathit{repair}(\mathbf{C}_l)$ and where $\mathbf{a}^{1:t_q-1}$ is a general sequence of action results, possibly including repair actions. We shall then prove that (17) holds for B^{t_r} where $t_r > t_q$, $\mathbf{a}^{t_q+1:t_r-1}$ consists of observations and operations only, and $a^{t_r} = \mathit{repair}(C_m)$.

If $\mathbf{a}^{t_q+1:t_r-1}$ consists of observations only, we have that $B_{ns}(\mathbf{e}^{1:t_r-1}) = B_{ns}(\mathbf{e}^{1:t_q})$ and that $B^{t_r-1} = B^{t_q}$, so in this case it is clear that (17) holds for B^{t_r-1}. If there is at least one operation action in $\mathbf{a}^{t_q+1:t_r-1}$ we know from the proof of Lemma 4 that the operation resets the BN. Furthermore, since there are no repairs in $\mathbf{a}^{t_q+1:t_r-1}$, we have that $B_{ns}(\mathbf{e}^{1:t_r-1}) = B_{ns}(\langle 0 \rangle)$ and that $B^{t_r-1} = B^0$. Again, it is clear that (17) holds for B^{t_r-1}.

Now, consider the last step, i.e. to update B^{t_r-1} to B^{t_r}. As in the proof of Lemma 5 we consider two cases. Case 1, where $O_j^{(t)}$ is such that $O_j^{(0)} \notin \mathit{de}_{B_{ns}}(C_i^{(0)})$; and Case 2, where $O_j^{(t)}$ is such that $O_j^{(0)} \in \mathit{de}_{B_{ns}}(C_i^{(0)})$.

Case 1. For this case, the computations are identical with Case 1 in the proof of Lemma 5.

Case 2. In Case 2 we consider variables $O_j^{(0)} \in de_{B_{ns}^{(0)}}(C_i^{(0)})$. Let $a^{t_{op}}$ be the last operation action before a^{t_r}. Then we have $t_q < t_{op} < t_r$ we have that $B_{ns}^{(t_r-1)} = B_{ns}^{(t_{op})} = B_{ns}^{(0)}$, and the computations in Case 2 in the proof of Lemma 5 are directly applicable. This ends the proof of Theorem 1. □

References

[Breese and Heckerman, 1996] Breese, J. S. and Heckerman, D. (1996). Decision-theoretic troubleshooting: A framework for repair and experiment. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*.

[Heckerman et al., 1995] Heckerman, D., Breese, J. S., and Rommelse, K. (1995). Decision-theoretic troubleshooting. *Communications of the ACM*, 38(3):49–57.

[Jensen and Nielsen, 2007] Jensen, F. V. and Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs*. Springer.

[Kipersztok and Wang, 2001] Kipersztok, O. and Wang, H. (2001). Another look at sensitivity of bayesian networks to imprecise probabilities. In *Proceedings of the 5th International Workshop on Artificial Intelligence and Statistics*.

[Langseth and Jensen, 2002] Langseth, H. and Jensen, F. V. (2002). Decision theoretic troubleshooting of coherent systems. *Reliability Engineering & System Safety*, 80(1):49–62.

[Lerner et al., 2000] Lerner, U., Parr, R., Koller, D., and Biswas, G. (2000). Bayesian Fault Detection and Diagnosis in Dynamic Systems. In *AAAI/IAAI*, pages 531–537.

[Martelli and Montanari, 1978] Martelli, A. and Montanari, U. (1978). Optimizing decision trees through heuristically guided search. *Commun. ACM*, 21(12):1025–1039.

[Murphy, 2002] Murphy, K. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, UC Berkeley, USA.

[Nilsson, 1980] Nilsson, N. J. (1980). *Principles of Artificial Intelligence*. Morgan Kaufmann, San Francisco, CA.

[Olive et al., 2003] Olive, X., Trave-Massuyes, L., and Poulard, H. (2003). AO* variant methods for automatic generation of near-optimal diagnosis trees. In *14th International Workshop on Principles of Diagnosis (DX 03)*, pages 169–174.

[Pearl, 2000] Pearl, J. (2000). *Causality*. Cambridge.

[Pernestål and Nyberg, 2009] Pernestål, A. and Nyberg, M. (2009). Non-stationary dynamic bayesian networks in modeling for troubleshooting. Submitted for publication to *International Journal of Approximate Reasoning*.

[Pernestål et al., 2009] Pernestål, A., Warnquist, H., and Nyberg, M. (2009). Modeling and troubleshooting with interventions. In *Proceedings of 2nd IFAC Workshop on Dependable Control of Discrete Systems (DCDS 02)*.

[Rintanen, 2004] Rintanen, J. (2004). Complexity of planning with partial observability. In *ICAPS 2004. Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling*, pages 345–354. AAAI Press.

[Robinson and Hartemink, 2008] Robinson, J. W. and Hartemink, A. J. (2008). Non-stationary dynamic bayesian networks. In *Proceedings of NIPS*.

[Russell and Norvig, 2003] Russell, S. and Norvig, P. (2003). *Artificial Intelligence. A Modern Approach*. Prentice Hall.

[Sun and Weld, 1993] Sun, Y. and Weld, D. S. (1993). A framework for model-based repair. In *In Proc. AAAI-93*, pages 182–187.

[Warnquist et al., 2009] Warnquist, H., Pernestål, A., and Nyberg, M. (2009). Anytime near-optimal troubleshooting applied to an auxiliary truck braking system. In *Proceedings of 6th IFAC Symposium on Fault Detection, Supervision and Safety of technical processes (SAFEPROCESS 2009)*.

Paper 5

A Comparison of Bayesian Approaches to Learning in Fault Isolation[1]

Anna Pernestål[*], Hannes Wettig[‡], Tomi Silander[‡], Mattias Nyberg[*], and Petri Myllymäki[‡]

[*]*Division of Vehicular Systems, Department of Electrical Engineering, Linköping University, Sweden.*

[‡]*Complex Systems Computations Group, Department of Computer Science, Helsinki Institute for Information Technology, Finland.*

Abstract

Fault isolation is the task of localizing faults in a process, given observations from it. To do this, a model describing the relations between faults and observations is needed. In this paper we focus on learning such models both from training data and from prior knowledge. There are several challenges in learning for fault isolation. The number of data and the available computing resources are often limited. Furthermore, there may be previously unobserved fault patterns. To meet these challenges we take on a Bayesian approach. We compare five different approaches to learning for fault isolation, and evaluate their performance on a real application, namely the diagnosis of an automotive engine.

[1]This paper has been submitted to Pattern Recognition Letters. It is based on [Pernestål et al., 2008]

1 Introduction

We consider fault isolation, i.e. the task of localizing faults that are present in a process given current observations from the process. To do this, a model of the relations between observations and faults is needed.

In many traditional methods for fault isolation, the model of the relations is given by knowledge about the physical behavior of process. It can for example be represented as a structure, a so called Fault Signature Matrix, describing which faults that may affect each observation, and possibly also how [Korbicz et al., 2004, Hamscher et al., 1992, Nyberg, 2005, Pulido et al., 2005]. We call such knowledge *expert knowledge*. In many applications there is, in addition to the expert knowledge, also data available from the process. This data can be used to learn about the process studied. In the current work we investigate and compare different methods for learning models of relations between faults and observations from training data. We also study the possibilities to integrate the training data with the expert knowledge.

The work is motivated by the problem of fault isolation in an automotive engine, and a Scania diesel engine is used as source for training and evaluation data. In engine fault isolation there may be several hundreds of possible faults and observations. There will typically be unobserved fault patterns, i.e. faults or combinations of faults from which there is no training data. Furthermore, training data is typically experimental, meaning that it is obtained by implementing faults, running the process, and collecting observations.

To meet the challenge of previously unobserved fault patterns we consider a Bayesian approach to learning for fault isolation. Within the Bayesian framework it is also possible to take other background information and expert knowledge into account, and not rely blindly on the data. We consider five different model classes when learning from training data. They are all previously presented in the literature in different forms. We tailor these methods to incorporate the available background information, and to become applicable to experimental data. The methods we consider are Direct Inference (DI), Logistic Regression (LogR), Linear Regression (LinR), Naive Bayes (NB) and general Bayesian Networks (BN).

The main contribution is the investigation of Bayesian learning methods for fault isolation by comparing models from the five classes mentioned above together with appropriate methods for learning their parameters. We do the comparison by application and evaluation of the methods using real-world data. In order to do the investigation of learning methods, we first discuss the characteristics of the fault isolation problem in terms of probability theory, and present performance measures that are meaningful for fault isolation. Thereafter we show how the five methods can be adopted to the isolation problem. We apply them to the task of fault isolation in the Scania diesel engine.

Bayesian methods for fault isolation have been previously studied in litera-

ture. In many of the previous works it is assumed that the model of the relations between faults and observations is given [Schwall and Gerdes, 2002, Lerner et al., 2000, Sheppard and Kaufman, 2005], or can be derived from a physical model without using training data [Narasimhan and Biswas, 2007, Roychoudhury et al., 2006], and focus is on *inference*. In the current work on the other hand, we consider five different model classes, and focus on *learning* the models of the relations, i.e. both structure and parameters.

Previous works on learning models, and in particular parameters in the models, for fault isolation from data rely on Bayesian methods as in [Weber et al., 2006], pattern recognition methods described for example in [Bishop, 2005, Devroye et al., 1996], or machine learning methods in [Heckerman et al., 1995b]. Applications are for example found in [Lee et al., 2007, Sheppard and Kaufman, 2005]. The methods in these previous works all rely on the fact that there is sufficient amount of training data available. Unfortunately, this is rarely the case in fault isolation, where the number of training samples often is limited, at least for rare and safety critical faults. Furthermore, there are often fault patterns from which there is no training data. The Bayesian approach to learning for fault isolation, used in the current paper, provides a sound method also in the case of missing data, and opens the possibility to take prior knowledge into account.

In [Pernestål and Nyberg, 2007], the problem of learning with missing fault patterns is discussed, and in [Pernestål and Nyberg, 2008] training data is combined with fundamental methods for fault isolation described in [de Kleer and Williams, 1992, Reiter, 1992]. The approach developed in [Pernestål and Nyberg, 2008] is referred to as DI in the current work, and compared to the other four methods for learning.

The paper is structured as follows. We introduce notation, and give a brief introduction to Bayesian networks in Section 2. We formulate the diagnosis problem in terms of probabilities in Section 3. Therein we also define relevant performance measures. In Section 4 we briefly describe the five methods used, and in particular how they are applied to the diagnosis problem. Then we perform evaluating experiments and compare the results obtained in Section 5. Finally, in Section 6 we conclude the paper by summarizing our results and discussing future work directions.

2 Preliminaries

Before going into the details of each of the learning methods we introduce notation that will be used, and give a brief introduction to Bayesian networks.

2.1 Notation

The fault isolation problem can be formulated as a filtering problem, where the task is to determine the fault(s) present in a process, given a set of observations from the process. Let the faults be represented by the binary variables $Y = (Y_1, \ldots, Y_K)$, where $Y_k = 1$ means that fault k is present, and let the observations be represented by the variables $\mathbf{X} = (X_1, \ldots, X_L)$, where each X_l is discrete or continuous. Generally, we use upper case letters to denote variables, and lower case letters to denote their values. Boldface letters denote vectors.

We write $p(\mathbf{x})$ to denote both probability distributions and probability density functions. The meaning will be clear from the context.

2.2 Fundamentals of Bayesian Networks

Bayesian networks are directed acyclic graphs in which nodes represent random variables and arcs represent directed probabilistic dependencies among the variables. We use the same notation for both nodes and variables, see for example [Jensen and Nielsen, 2007]. A Bayesian network encodes the joint probability distribution over a finite set of variables $\{W_1, \ldots, W_M\}$, and decomposes it into a sequence of conditional probability distributions, one for each variable.

More specifically, let $pa(W_i)$ denote the parents of W_i, and let $pa(W_i)$ be a value (configuration) of $pa(W_i)$. Then there is a conditional probability distribution $p(w_i|pa(w_i))$ for each variable W_i. Nodes without parents are called *root nodes*. The conditional probability of a root node W_i is simply its prior probability $p(w_i)$. The joint probability distribution of the variable set $\{W_1, \ldots, W_L\}$ can be obtained by taking the product of all these conditional probability distributions:

$$p(w_1, \ldots, w_M) = \prod_{i=1}^{L} p(w_i|pa(w_i)). \tag{1}$$

In Bayesian networks, both the presence of arcs, and their directions, as well as the absence of arcs encodes knowledge about dependencies and independences. In addition to the structure of dependencies characterized by the arcs in the Bayesian network, it also includes all the distributions $p(w_i|pa(w_i))$. When we discuss learning in Bayesian networks, we mean learning both the structure and the probability distributions. More details about Bayesian networks can be found for example in [Jensen and Nielsen, 2007] and [Russell and Norvig, 2003]

3 Bayesian Fault Isolation

We are now ready to formulate the problem of performing fault isolation in a process in probabilistic terms, and to present relevant performance measures.

3.1 Problem Formulation

In addition to the current observation $\mathbf{X}$ from the process under diagnosis, a set of training data $\mathcal{D}$ is given. Training data consists of samples $(\mathbf{y}^n, \mathbf{x}^n)$, $n = 1, \ldots, N_{\mathcal{D}}$, of pairs of fault and observation variables. The training data is collected by implementing faults in the process, and then collecting observations. This means that training data is *experimental*. To evaluate the fault isolation methods we use an evaluation set $\mathcal{E}$, consisting of $N_{\mathcal{E}}$ samples. The evaluation data is collected by running the process without intervening with it, i.e. without implementing any faults but rather observing faults as they appear. Thus, evaluation data is *observational*.

We assume that the fault isolation algorithm is triggered by a fault detector telling us that there must be *at least one fault present* in the process.

The structure of dependencies between the faults and observations has three basic properties, illustrated in the example Bayesian network of Figure 1. The first property is that faults are assumed to be a priori independent, i.e. that

$$p(\mathbf{y}) = \prod_{k=1}^{K} p(y_k|y_1, \ldots, y_{k-1}) \approx \prod_{k=1}^{M} p(y_k), \tag{2}$$

meaning that faults do not cause other faults to occur. Although not necessary for the methods in the current work, this is a standard assumption in many fault isolation algorithms [Hamscher et al., 1992], and it simplifies the reasoning in the following sections.

Second, faults may causally affect one or several of the observation variables introducing dependencies between faults and variables. A dependency between fault variable Y_k and observation variable X_l means that the fault *may* be visible in the observation.

The third property is that an observation variable X_l may be dependent on other observation variables. Dependencies between observation variables can arise due to several reasons. For example they can be caused by unobserved and unmodeled factors, such as the surroundings of the automotive engine, the behavior of the driver, and the operation point of the engine. These unobserved factors could be modeled using hidden nodes, but since they are numerous and their explicit effects are unknown they are here approximated with direct dependencies between observation variables. This is more carefully discussed in [Pernestål et al., 2006].

In the current work we take a Bayesian view on fault isolation. The objective is to find the probability that each fault is present given the current observation, the training data, and the prior knowledge $\mathbf{i}$, i.e. to compute the probabilities $p(y_k|\mathbf{x}, \mathcal{D}, \mathbf{i})$, $k = 1, \ldots, K$. The probability for a fault y_k can be found by

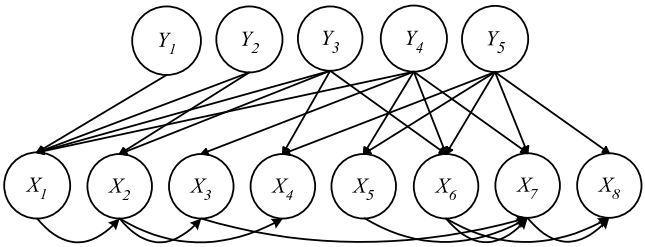


Figure 1: A Bayesian network describing a typical fault isolation problem.

marginalizing over all other faults $y_{\bar{k}} = (y_1, \ldots, y_{k-1}, y_{k+1}, \ldots, y_K)$,

$$p(y_k|\mathbf{x}, \mathcal{D}, \mathbf{i}) = \sum_{y_{\bar{k}}} p(y_{\bar{k}}, y_k|\mathbf{x}, \mathcal{D}, \mathbf{i}). \tag{3}$$

Note that $(y_{\bar{k}}, y_k) = \mathbf{y}$, and (3) means that we search the conditional distribution $p(\mathbf{y}|\mathbf{x}, \mathcal{D}, \mathbf{i})$. To simplify the notation we will not write out the prior knowledge $\mathbf{i}$ explicitly in the equations.

Computing the conditional distribution $p(\mathbf{y}|\mathbf{x}, \mathcal{D})$ of $\mathbf{y}$ directly from data $\mathcal{D}$ is generally difficult. Instead, we approximate it using a model $\mathcal{M}(\mathcal{D})$ learned from data, i.e.

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}) \approx p(\mathbf{y}|\mathbf{x}, \mathcal{M}(\mathcal{D})) = p_{\mathcal{M}(\mathcal{D})}(\mathbf{y}|\mathbf{x}), \tag{4}$$

where we have introduced the notation $p_{\mathcal{M}(\mathcal{D})}(\mathbf{y}|\mathbf{x})$ to denote the distribution obtained from training data $\mathcal{D}$ by using model $\mathcal{M}$ and the parameters determined using the appropriate method. To simplify notation we write $p_{\mathcal{M}}(\mathbf{y}|\mathbf{x})$ when there is no risk for confusion which data that is used.

The model $\mathcal{M}(\mathcal{D})$ can for example be a Bayesian network or a regression model. Methods for learning the parameters of different types of models will be discussed in Section 4.

3.2 Performance Measures

To evaluate the different models to be used in Bayesian fault isolation, we use two performance measures: the logistic score and the percentage of correct classification.

The logistic score is a commonly used performance measure [Bishop, 2005, Mitchell, 1997]. The logistic score is based on a set $\mathcal{E}$ of evaluation data and is given by

$$\mu(\mathcal{E}, \mathcal{M}) = \frac{1}{N_{\mathcal{E}}} \sum_{n=1}^{N_{\mathcal{E}}} \log p_{\mathcal{M}(\mathcal{D})}(\mathbf{y}^n|\mathbf{x}^n). \tag{5}$$

The score μ measures two important properties of the fault isolation system: the ability to assign large probability mass to faults that are present, as well as the ability to assign small probability mass to faults that are not present. Furthermore, the log-score is a *proper score*. A proper score has the characteristic that it is maximized when the learned probability distribution is equal to the generating distribution. In fault isolation applications, the conditional probabilities of faults can be combined with decision theoretic methods to determine the appropriate counter-action, see for example [Heckerman et al., 1995a, Langseth and Jensen, 2002, Pernestål and Nyberg, 2007]. In decision theory, optimal decision making requires conditional probabilities close to the generating distribution, and thus proper scores are suitable.

The second performance measure we use, percentage of correct classification, is not a proper scoring function. However, it is closely related to the 0/1-loss used for example in pattern classification [Bishop, 2005]. We define

$$\nu(\mathcal{E}, \mathcal{M}) = \frac{|\mathcal{C}|}{N_{\mathcal{E}}}, \tag{6}$$
$$\text{where} \quad \mathcal{C} = \{n : y_k^n = 1, k = \arg\max_{k'} p_{\mathcal{M}(\mathcal{D})}(y_{k'}|\mathbf{x}^n)\},$$

and y_k^n denotes element k in $\mathbf{y}^n$. In words, $\mathcal{C}$ is the set of all indicies where the underlying fault is assigned the largest probability when model $\mathcal{M}$ is used, and the ν-score is thus the fraction of cases in evaluation data where the underlying fault is correctly classified. In case of multiple faults present it suffices to assign highest probability to any of them. The ν-score reflects the performance of the fault isolation system combined with the simple troubleshooting strategy "check the most probable fault first".

4 Modeling Methods

In this section we briefly present the modeling methods used, i.e. the different models used and methods for determining the parameters therein. We carefully state all assumptions made, and describe the adjustments of each method to apply it to the isolation problem. However, we begin by describing two assumptions that need to be made for all methods except DI.

4.1 Modeling Assumptions

In all the methods considered in this paper – with the exception of DI – separate models are built for each fault, and thus independence between the faults is assumed. This approach is illustrated in Figure 2. Before any training data is recorded, this assumption corresponds to (2). Since faults are inflicted in training data, the data does not include any information about co-occurrence of the faults, such as how faults affect each other or whether there are unknown

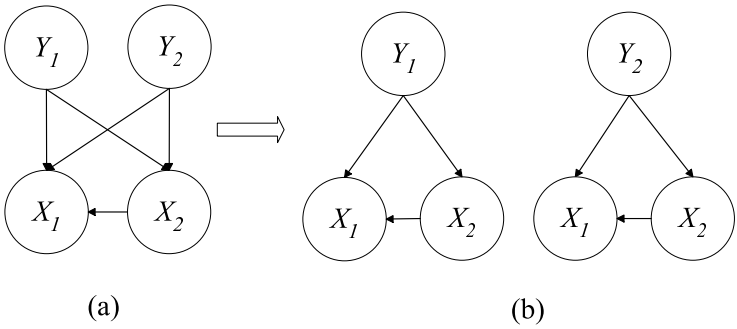


Figure 2: The principle of approximating one model of all faults with one for each fault.

effects that cause certain faults to appear at the same time. This fact is partly handled by building a separate model for each fault. However, building separate models, also induce a stronger assumption, namely that the faults *remain* independent given the observations:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{k=1}^{K} p(y_k|\mathbf{x}, y_1, \ldots, y_{k-1}) \approx \prod_{k=1}^{K} p(y_k|\mathbf{x}) \tag{7}$$

By applying Bayes' rule on the first probability in (7) we have

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})}. \tag{8}$$

Bayes' rule on the last probability in (7) gives

$$\prod_{k=1}^{K} p(y_k|\mathbf{x}) = \prod_{k=1}^{K} \frac{p(\mathbf{x}|y_k)p(y_k)}{p(\mathbf{x})} = \frac{p(\mathbf{y})\prod_{k=1}^{K} p(\mathbf{x}|y_k)}{p(\mathbf{x})}, \tag{9}$$

where we have used (2) in the last step. By approximation (7) the expressions in (8) and (9) are equal. Thus, the approximation (7) is equivalent to

$$p(\mathbf{x}|y_1, \ldots, y_K) \approx \frac{1}{p(\mathbf{x})^{K-1}} \prod_{k=1}^{K} p(\mathbf{x}|y_k) \tag{10}$$

In (10) $p(\mathbf{x})$ is a normalization constant, and the equation means that the observation $\mathbf{x}$ is dependent on each fault y_k, but this dependency is assumed to be independent of all other faults $y_{k'}, k' \neq k$. In other words, we assume no *"explaining away"* effect [Jensen and Nielsen, 2007]. The explaining away effect can be understood as follows. Consider Bayesian network with two faults Y_1 and Y_2 and two observations, where X_1 that is dependent on both faults

and X_2 is dependent on Y_2 only. Assume that observation X_1 indicates that there is a fault present (we say that X_1 "alarms"). Then both faults Y_1 and Y_2 are potential explanations. Now, assume that we learn that fault Y_2 is present (for example by observing that X_2 alarms), then fault Y_2 is likely to be the explanation of the alarm X_1 also. Since X_1 is explained by fault Y_2, fault Y_1 becomes less probable. The presence of Y_2 have *explained away* Y_1 through the observation X_1.

The explaining away effect occurs when there are unshielded colliders, i.e. common children of two or more nodes which are them-self not connected. Looking at Figure 1 we observe ignoring explaining away is indeed is a strong assumption, since there are several unshielded colliders of the faults. However, since each fault is allowed to be dependent on all observations, the explaining away effect will be partially encoded in the direct dependencies between faults and observations.

Assumption (7) is primarily made for technical reasons, in order to be able to build separate models for each fault. However, it is often the case (as in the application in Section 5) that there is training data only from single faults. Using training data straight-forwardly, this would lead to that we learn a strong dependence between the faults: if one fault is present, other faults are not. By approximation (7) this is avoided, and we do not learn these dependencies.

From Section 2 we know that it is assumed that there is at least one fault present. Recall that $\sum_k y_k > 0$ means that there is at least one fault present, and, similarly, that $\sum_k y_k = 0$ means that there is no fault present. The knowledge that there is at least one fault present introduce dependencies between the single fault models in (7), since in general we have

$$p(\mathbf{y}|\mathbf{x}, \sum_k y_k) \neq \prod_k p(y_k|\mathbf{x}, \sum_k y_k > 0). \tag{11}$$

To avoid this recoupling between models, we study the probability for the faults given the knowledge that at least one fault is present i detail. We have

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}, \sum_k y_k > 0, \mathcal{D}) &= \frac{p(\sum_k y_k > 0|\mathbf{y}, \mathbf{x}, \mathcal{D}) p(\mathbf{y}|\mathbf{x}, \mathcal{D})}{p(\sum_k y_k > 0|\mathbf{x}, \mathcal{D})} = \\ &= \begin{cases} 0 & \sum_k y_k = 0, \\ \frac{p(\mathbf{y}|\mathbf{x}, \mathcal{D})}{1 - p(\mathbf{Y} = 0|\mathbf{x}, \mathcal{D})}, & \sum_k y_k \neq 0. \end{cases} \end{aligned} \tag{12}$$

In the current paper we ignore the fact that at least one fault is present during the learning phase and the single-fault models are trained individually. We then apply (12) in the evaluation phase.

Table 1: An example of an FSM

	Y_1	Y_2	Y_3
X_1	$\mathcal{X}$	$\mathcal{X}$	0
X_2	$\mathcal{X}$	0	$\mathcal{X}$

4.2 Direct Inference

The first method for fault isolation that we present is Direct Inference (DI). Similar to several previous fault isolation algorithms, DI relies on prior knowledge about which observations that may be affected by each fault [de Kleer and Williams, 1992, Reiter, 1992, Korbicz et al., 2004]. Such information is typically expressed in a so called Fault Signature Matrix (FSM). An example of an FSM is given in Table 1. In the FSM, a zero in position (l, k) means that fault Y_k can never affect observation X_l, while a $\mathcal{X}$ mean that Y_k *may* affect observation X_l. DI aims at combining the information from the FSM with the training data available. Assuming that observations are binary and that the background information $\mathbf{i}$ contains the FSM. Then, under certain assumptions it can be shown [Mitchell, 1997, Pernestål and Nyberg, 2007] that

$$p_{\mathrm{DI}(\mathcal{D})}(\mathbf{y}|\mathbf{x}, \alpha_{\mathbf{x}y}) = \begin{cases} 0 & \mathbf{x} \in \gamma \\ \frac{n_{\mathbf{x}y}+\alpha_{\mathbf{x}y}}{N_y+A_y} \frac{p(\mathbf{y}|\mathbf{i})}{\pi_0} & \text{otherwise,} \end{cases} \tag{13}$$

where π_0 is a normalization constant, $n_{\mathbf{x}y}$ is the count in training data $\mathcal{D}$ where the fault is $\mathbf{y}$ and the observation is $\mathbf{x}$, $\alpha_{\mathbf{x}y}$ is a parameter describing the prior belief in the observation $\mathbf{x}$ when the fault is $\mathbf{y}$. The parameters α can be seen as hypothetical samples, which would have been obtained if our prior beliefs were true. The parameters α are sometimes referred to as *Dirichlet* parameters, since a Dirichlet prior is used in the computations. Furthermore, $N_y = \sum_{\mathbf{x}'} n_{\mathbf{x}'\mathbf{y}}$ and $A_y = \sum_{\mathbf{x}'} \alpha_{\mathbf{x}'\mathbf{y}}$. The set γ is determined from the fact that some observations are impossible according to the FSM as described in [Pernestål and Nyberg, 2008].

The DI method has been developed for sparse sets of training data, particularly when there is only training data from a subset of the fault patterns to isolate.

4.3 Bayesian Network Methods

When using Bayesian networks for filtering, the Bayesian network of the joint distribution $p(y, \mathbf{x}|\theta)$ is modeled. Here, θ are parameters in the conditional probability distributions associated with the nodes in the network, see Section 2.2. From the joint distribution, the conditional distribution for each of the faults y_k can be computed. As described in Section 4, one model for each fault is

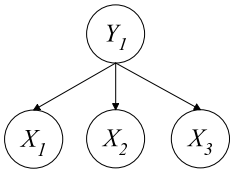


Figure 3: Naive Bayes network structure.

built. The models are combined by using (7) and correcting for the knowledge that there is at least one fault present by using (12). The probability for each fault is then determined by marginalization. We consider two types of Bayesian networks: Naive Bayes (NB) and general Bayesian Networks (BN).

Naive Bayes

In a *Naive Bayes* network it is assumed that the observations are independent given the fault. This structure is exemplified in Figure 3. We assume this structure, and learn the parameters in the conditional probabilities using standard methods described for example in [Heckerman et al., 1995b]. Naive Bayes is one of the most commonly used methods for Bayesian prediction and often performs surprisingly well [Devroye et al., 1996, Rish, 2001]. However, if there are strong dependencies between observations, the independence assumption made may introduce unnecessary large errors. For example, assume that two observations are identical. In this case, a better inference result may be obtained ignoring some of the observations that are strongly dependent. To alleviate this problem, we apply a variable selection according to an internal leave-one-out scoring function. This approach was first introduced in [Langley and Sage, 1994], where it is called *selective* naive Bayes classifier. Let $\mathcal{V} = 2^{\mathbf{X}}$ be the set of all subsets of the observations, let $V \in \mathcal{V}$, and let $\mathcal{N}_V$ be the Naive Bayesian network defined by V. We then choose the variable set V^* according to

$$V^* = \arg\max_{V \in \mathcal{V}} = \frac{1}{N_{\mathcal{D}}} \sum_{n=1}^{N_{\mathcal{D}}} \log p_{\mathcal{N}_V(\mathcal{D} \setminus \{(\mathbf{y}^n, \mathbf{x}^n)\})}(y_k^n | \mathbf{x}^n, \alpha),$$

where α is the Dirichlet hyper-parameter for the NB model, and are tuning parameters. The probabilities for fault y_k is computed by

$$p_{\mathcal{N}_{V^*}(\mathcal{D})}(y_k | \mathbf{x}, \alpha).$$

General Bayesian Network

A natural extension of the naive Bayes model is to allow a more general structure for each fault, and learn both structure and conditional probabilities from

the training data. However, it is known that the faults causally precede the observations. Therefore we restrict the possible structures to the ones where the fault node is a root node. This is the only constraint used. One Bayesian network (BN) was learned for each fault using a BDe score with an equivalent sample size parameter of 1.0 [Heckerman et al., 1995b]. For small systems (< 30 variables) learning can be performed using the exact algorithm in [Silander and Myllymäki, 2006], while for larger systems approximate methods, e.g. [Heckerman et al., 1995b, Mitchell, 1997, Russell and Norvig, 2003], can be used.

Let $\mathcal{B}$ denote the Bayesian network learned using the BDe score. Then the probabilities for fault y_k is computed by

$$p_{\mathcal{B}(\mathcal{D})}(y_k|\mathbf{x}, \alpha), \tag{14}$$

where α is again the Dirichlet hyper-parameter.

4.4 Regression

Fault isolation is a discriminative task, where we are to predict the fault vector $\mathbf{y}$ given the observations $\mathbf{x}$, i.e. to estimate the conditional probability of $\mathbf{y}$

$$p(\mathbf{y}|\mathbf{x}, \theta) = \frac{p(\mathbf{y}, \mathbf{x}|\theta)}{\sum_{\mathbf{y}} p(\mathbf{y}, \mathbf{x}|\theta)}. \tag{15}$$

It is well known [Ng and Jordan, 2002, Kontkanen et al., 2001, Friedman et al., 1997] that in such a case it can be of great benefit to employ a discriminative learning method, that only learns the probabilities asked, instead of wasting training data to learn the joint data likelihood as in the Bayesian network methods of Section 4.3. Regression models form a family of such methods, and here we consider two classes of such: linear and logistic regression models.

In the previous methods, as well as in training data, the variable y_k representing the faults is a discrete variable, but in the computations in the regression methods, we relax it to be a continuous variable.

Linear Regression

The most straight-forward regression method is linear regression, where each fault variable is assumed to be a linear combination of the observations plus a gaussian noise term,

$$y_k = \mathbf{w}_k^T \mathbf{x} + w_{k0} + \epsilon_k, \quad \epsilon_k \sim N(0, \sigma_k).$$

Here $\mathbf{w}_k$, w_{k0}, and σ_k are parameters to be determined. This gives the probability distribution

$$p_{\mathrm{LinR}}(y_k|\mathbf{x}) = \frac{1}{Z} \exp(-\frac{(\mathbf{w}_k^T \mathbf{x} + w_{k0} - y_k)^2}{2\sigma_k^2}),$$

where Z is a normalization constant. To determine the parameters we use the standard methods described for example in [Bishop, 2005]. For example,

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{n=1}^{N_{\mathcal{D}}} (\mathbf{w}_k^T \mathbf{x}^n + w_{k0} - y_k^n)^2.$$

When the parameters $\mathbf{w}^*$ are known, the parameters σ and Z can also be computed [Bishop, 2005].

Logistic Regression

Learning parameters to maximize (15) for a Bayesian network is known to be equivalent to *logistic regression* under the condition that no node can be a "bastard", i.e. a common child of two variables that are not directly interconnected them selfs. More formal definition and proofs can be found in [Roos et al., 2005]. In our case, this fact is guaranteed by assumption (7).

To start with, for each fault we learn a logistic regression model corresponding to a discriminative Naive Bayes classifier [2]. Let α and β be parameters in the logistic regression model, and define

$$p_{\mathrm{LogR}}(Y_k = 1|\mathbf{x}, \alpha, \beta) = \frac{\exp s(\mathbf{x}, \alpha, \beta)}{\exp s(\mathbf{x}, \alpha, \beta) + \exp -s(\mathbf{x}, \alpha, \beta)}$$
$$\text{where} \quad s(\mathbf{x}, \alpha, \beta) = \alpha + \sum_{l=1}^{L} x_l \beta_l.$$

When learning the parameters α and β, we use a smoothing term $c(\alpha, \beta)$ in the objective function. The smoothing function takes the place of a prior probability distribution for the parameters. To determine the smoothing term, we normalize training data such that

$$\sum_n x_l^n = 0 \quad \text{and} \quad \max_n |x_l^n| = 1$$

Then, beginning with a uniform prior, c', we pretend to have seen one vector of each fault at node Y_k and two vectors of each fault with extreme values ± 1 at each node X_l, with all other values unobserved. This amounts to a smoothing term

$$c'' = c' - 2\log(\exp(\alpha) + \exp(-\alpha)) - \\ -4\sum_{l=1}^{L} \log(\exp(\beta_l) + \exp(-\beta_l)).$$

[2] Possible other choices include tree-augmented Naive Bayes (TAN) [Friedman et al., 1997, Roos et al., 2005, Greiner and Zhou, 2002].

This smoothing term is problematic since it is flat near zero, leading to that no parameters will be exactly zero. In logistic regression many small parameters can make a large difference in the inference result, while they may be weakly supported. To avoid the flatness around zero $\log(\exp(z)+\exp(-z))$ was replaced by $|z|$ to obtain c from c''. This is a good approximation away from zero, but forces unsupported parameters to zero, implicitly performing attribute selection.

For fault y_k we search parameters that maximize

$$\log p_{\mathrm{LogR}}(y_k|\mathbf{x},\alpha,\beta) + c(\alpha,\beta) = \\ = \sum_{n=1}^{N_{\mathcal{D}}} \log p(y_k^n|\mathbf{x}^n,\alpha,\beta) - 2|\alpha| - 4\sum_{l=1}^{L}|\beta_l|.$$

We do this by simple line search, one parameter at a time[3].

Finally, we apply also a variant of LogR, which we denote "LogR + weights", where training vectors are weighted according to their prior probabilities $p(y_k)$. This is done since the training data and the evaluation data are known to have different distributions. The idea is to weight the training vectors in the objective function as to focus the optimization on areas of the data space more likely to be seen later on. The corresponding objective function for fault Y_k becomes

$$\sum_{n=1}^{N_{\mathcal{D}}} \log w_k p(y_k^n|\mathbf{x}^n,\alpha,\beta) + c(\alpha,\beta). \tag{16}$$

where the weight w_k is the prior $p(y_k)$ divided by the observed relative frequency $\#\{n : y_k^n = y_k\}/N_{\mathcal{D}}$.

5 Experiments

To evaluate the different modeling methods for fault isolation, we apply them to the diagnosis of the gas flow in a 6-cylinder diesel engine in a Scania truck. In automotive engines, sensor faults are one of the most common faults, and here we consider five faults that may appear in different sensors. The faults are listed together with their prior probabilities for single faults in Table 2. Note that the probabilities in Table 2 do not sum to one, since the probabilities for multiple faults are not included.

5.1 Experimental Setup

For the gas flow of the diesel engine there is a physical model from which a set of 29 residuals are automatically generated using structural analysis [Svärd and

[3]For larger problems faster methods, as for example discussed in [Minka, 2003] could be more suitable.

Table 2: The faults considered

Fault	description	$p(y_k)$
y_1	exhaust gas pressure	0.4
y_2	intake pressure	0.13
y_3	intake air pressure	0.057
y_4	EGR vault position	0.13
y_5	mass flow	0.057

Nyberg, 2009, Krysander et al., 2008]. The residuals, which are constructed to be sensitive to subsets of the faults, are used as observations in the fault isolation.

For training and evaluation data we use measurements from real operation of the truck, with faults implemented. The training data consists of 100 samples each from the five single faults. Evaluation data consists of data from the five single faults, but also of data from two multiple faults $y_1\&y_2$, and $y_1\&y_4$. Evaluation data is observational, and consists of 1000 samples, distributed roughly according to the prior probabilities in Table 2.

The data we consider is originally continuous, but generally not Gaussian distributed. All methods, except the two regression algorithms, take in discrete data. The data is discretized in two different ways: binary, with thresholds set such that all fault free data in the training set is contained in the same bin; and discretized using k-means clustering [Hartigan, 1975] with $k = 4$. DI is applied to the discrete data. NB and BN are run both on discrete and binary data. The regression methods LinR and LogR are applied to the continuous data. To learn the BN model, we use the exact algorithm in [Silander and Myllymäki, 2006], and in DI, BN, and NB we set the Dirichlet parameters to 1.

As described in Section 4 the NB and DI methods perform best if not all observations are used. For both DI and NB we perform variable selection such that an internal logistic score is maximized. For DI, the best result is obtained by using only six of the observations. In NB between seven and 18 observations are used for each fault.

5.2 Results

In Table 3 the logistic score (μ) and percentage of correct classification (ν) are presented for the different methods. In addition we report the number of parameters used by each predictor. This is relevant, since for on-board fault isolation the computing and storage capacity is often limited. For comparison we also report the default which is obtained by simply using the prior probabilities given in Table 2.

Table 3: Comparison of the methods

method	μ-score	ν-score	#pars
DI	-1.088	0.781	106
NB-bin.	-1.340	0.748	293
NB-disc.	-1.044	0.843	335
BN-bin.	-1.297	0.782	287
BN-disc.	-1.398	**0.840**	1136
LinR	-1.839	0.834	150
LogR	-1.071	0.829	46
LogR+weights	**-0.953**	0.829	44
default	-1.738	0.592	5

Considering the μ-score, we see that among the four best methods in Table 3 three are discriminative and learn the conditional distribution instead of the joint distribution. Furthermore, LogR with training sample weighting performs best on this data in logistic score sense, while using a small number of parameters. Surprisingly the weighting trick has made quite a difference and LogR without weights it is outperformed by NB-disc. NB performs better when it is fed with discretized observations instead of binary, while for BN the effect is reversed. Clearly the discretized data contains more information, but it seems that in more complex Bayesian networks the conditional probability tables grow too large, and there is not enough training data to learn them accurately. In DI good results are obtained by exploiting prior FSM knowledge in terms of that some faults never cause an observation to pass certain thresholds.

Measured by the ν-score the relative differences between the methods is smaller. This score favors the regression models and the Bayesian methods using discrete data.

To exemplify the results, Table 4 compares the logistic scores of the predictions given for the single faults by DI and LogR+weights. Note that because of the inequality (11) the columns do not sum to the corresponding entries in Table 3. Both methods (as all others) have most trouble with isolating faults y_1, y_2 and y_4, the ones appearing simultaneously in evaluation data, but not in training data. This gives evidence for explaining away being important in this problem. Figure 4, in which the probabilities for each fault using LogR + weights are plotted, shows this in more detail. In the figure, the true fault is marked with a solid line. Moreover, we have ordered the evaluation data such that the right-most samples have multiple faults, visualizing that the double faults are most difficult to predict.

Table 4: Comparision of DI and LogR on single faults

fault	μ DI	μ LogR+w
y_1	-0.346	-0.385
y_2	-0.324	-0.287
y_3	-0.087	-0.008
y_4	-0.334	-0.294
y_5	-0.177	-0.133

6 Conclusions

We have considered the problem of fault isolation in an automotive diesel engine, and discussed the special characteristics of this problem. There is experimental training data available which is distributed differently from the observational data to which the diagnosis algorithm is to be applied. In particular, evaluation data consists partly of previously unseen fault patterns. In addition there is prior knowledge available about which faults that may affect each observation, and also the knowledge that at least one fault is present.

We have studied different Bayesian and regression approaches to combine this by-nature heterogeneous information into probability distributions for the faults conditioned on given observations. We have compared the performance of the methods using real-world data, and have found that on the application studied the discriminative logistic regression method performs best. Among the methods that perform well we have also found the naive Bayes classifier and the direct inference method.

One of the clearest implications of this work is that all methods have difficulties with handling unobserved fault patterns. Unfortunately, unobserved patterns are common in fault isolation, so this problem should be tackled in future work. The four methods where one model is build for each fault, let the explaining-away effect be present only through observations. However, this explaining-away effect can possibly be helpful when diagnosing unseen patterns. DI performs among the best of the methods, despite its few parameters. DI is also the only method that include background information in the learning phase, we therefor believe that the it is crucial to utilize background information whenever it is available, in particular when there are unseen patterns.

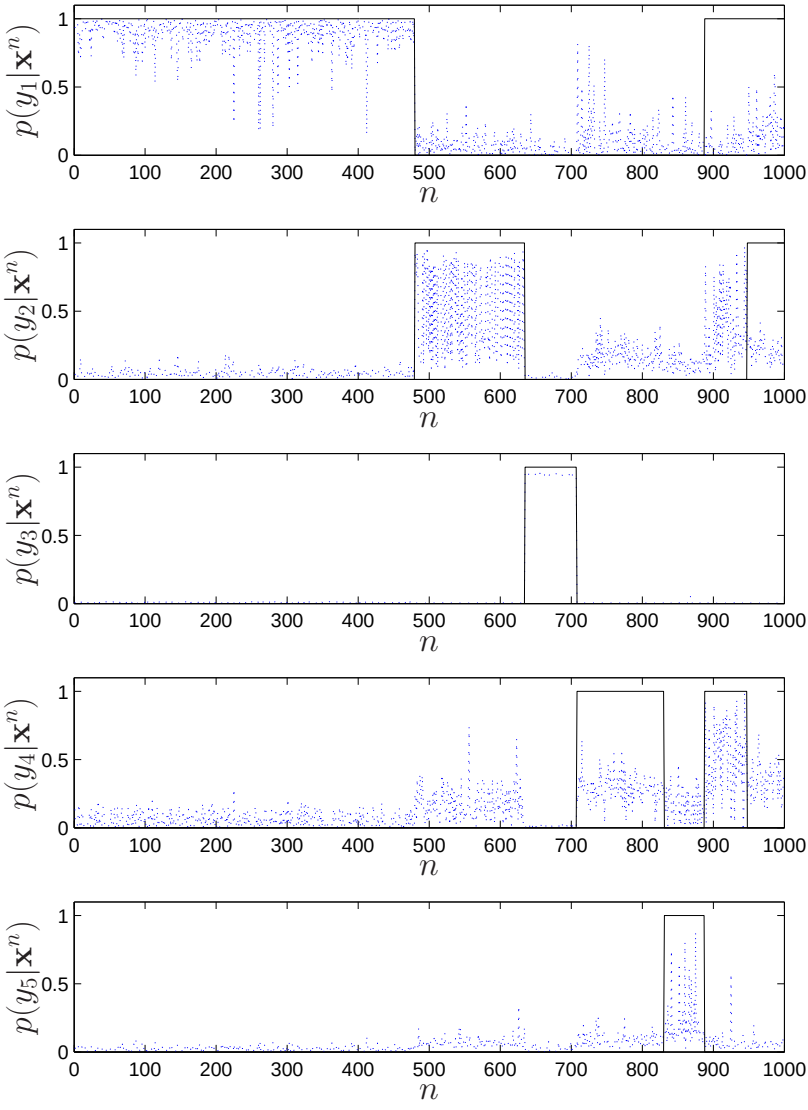


Figure 4: The predicted probability for the different faults given by LogR+w. Evaluation data is ordered after their fault patterns. The true fault is marked with a solid line.

References

[Bishop, 2005] Bishop, C. M. (2005). *Neural Networks*. Oxford University Press.

[de Kleer and Williams, 1992] de Kleer, J. and Williams, B. C. (1992). Diagnosis with Behavioral Modes. In *Readings in Model-based Diagnosis*, pages 124–130, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Devroye et al., 1996] Devroye, L., Györfi, L., and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer, New York.

[Friedman et al., 1997] Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning*, pages 131–163.

[Greiner and Zhou, 2002] Greiner, R. and Zhou, W. (2002). Structural Extension to Logistic Regression: Discriminative Parameter Learning of Belief Net Classifiers. In *13th international conference on uncertainty in artificial intelligence*.

[Hamscher et al., 1992] Hamscher, W., Console, L., and deKleer, J. (1992). *Readings in Model-based Diagnosis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Hartigan, 1975] Hartigan, J. A. (1975). *Clustering Algorithms*. Wiley.

[Heckerman et al., 1995a] Heckerman, D., Breese, J. S., and Rommelse, K. (1995a). Decision-theoretic troubleshooting. *Communications of the ACM*, 38(3):49–57.

[Heckerman et al., 1995b] Heckerman, D., Geiger, D., and Chickering, D. M. (1995b). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20(3):197–243.

[Jensen and Nielsen, 2007] Jensen, F. V. and Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs*. Springer.

[Kontkanen et al., 2001] Kontkanen, P., Myllymäki, P., and Tirri, H. (2001). Classifier learning with supervised marginal likelihood. In Breese, J. and Koller, D., editors, *Proceedings of the 17th International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 277–284.

[Korbicz et al., 2004] Korbicz, J., Koscielny, J. M., Kowalczuk, Z., and Cholewa, W. (2004). *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Springer, Berlin, Germany.

[Krysander et al., 2008] Krysander, M., Åslund, J., and Nyberg, M. (2008). An Efficient Algorithm for Finding Minimal Over-constrained Sub-systems for Model-based Diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 38(1):197–206.

[Langley and Sage, 1994] Langley, P. and Sage, S. (1994). Induction of Selective Bayesian Classifiers. In *Proceedings of the 10th Conference on Uncertainty Artificial Intelligence*.

[Langseth and Jensen, 2002] Langseth, H. and Jensen, F. V. (2002). Decision theoretic troubleshooting of coherent systems. *Reliability Engineering & System Safety*, 80(1):49–62.

[Lee et al., 2007] Lee, G., Bahri, P., Shastri, S., and Zaknich, A. (2007). A multi-category decision support framework for the tennessee eastman problem. In *Proceedings of the European Control Conference 2007*, Greece.

[Lerner et al., 2000] Lerner, U., Parr, R., Koller, D., and Biswas, G. (2000). Bayesian Fault Detection and Diagnosis in Dynamic Systems. In *AAAI/IAAI*, pages 531–537.

[Minka, 2003] Minka, T. P. (2003). A comparison of numerical optimizers for logistic regression. Technical report, Micrsoft Research.

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

[Narasimhan and Biswas, 2007] Narasimhan, S. and Biswas, G. (2007). Model-Based Diagnosis of Hybrid Systems. *IEEE Transactions on Man, Systems and Cybernetics – part A*, 37(3):348–361.

[Ng and Jordan, 2002] Ng, A. Y. and Jordan, M. I. (2002). On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems 14*.

[Nyberg, 2005] Nyberg, M. (2005). Model-Based Diagnosis of an Automotive Engine Using Several Types of Fault Models. *IEEE Transactions on Control Systems Technology*, 10(5):679–689.

[Pernestål and Nyberg, 2007] Pernestål, A. and Nyberg, M. (2007). Probabilistic Fault Isolation Based on Incomplete Training Data with Application to an Automotive Engine. In *Proceedings of the European Control Conference (ECC 07)*.

[Pernestål and Nyberg, 2008] Pernestål, A. and Nyberg, M. (2008). Bayesian Fault Isolation by Combining Data and Process Knowledge with Application to Engine Diagnosis. submitted to *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*.

[Pernestål et al., 2006] Pernestål, A., Nyberg, M., and Wahlberg, B. (2006). A Bayesian Approach to Fault Isolation with Application to Diesel Engine Diagnosis. In *Proceedings of 17th International Workshop on Principles of Diagnosis (DX 06)*, pages 211–218.

[Pernestål et al., 2008] Pernestål, A., Wettig, H., Silander, T., Nyberg, M., and Myllymäki, P. (2008). A bayesian approach to learning in fault isolation. In *Proceedings of the 19th International Workshop on Principles of Diagnosis*.

[Pulido et al., 2005] Pulido, B., Puig, V., Escobet, T., and Quevedo, J. (2005). A New Fault Localization Algorithm that Improves the Integration Between Fault Detection and Localization in Dynamic Systems. In *Proceedings of 16th International Workshop on Principles of Diagnosis (DX 05)*.

[Reiter, 1992] Reiter, R. (1992). A Theory of Diagnosis From First Principles. In *Readings in Model-based Diagnosis*, pages 29–48, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Rish, 2001] Rish, I. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*.

[Roos et al., 2005] Roos, T., Wettig, H., Grünwald, P., Myllymäki, P., and Tirri, H. (2005). On Discriminative Bayesian Network Classifiers and Logistic Regression. *Machine Learning*, pages 267–296.

[Roychoudhury et al., 2006] Roychoudhury, I., Biswas, G., and Koutsoukos, X. (2006). A Bayesian Approach to Efficient Diagnosis of Incipient Faults. In *17th International Workshop on Principles of Diagnosis (DX 06)*, pages 243–250.

[Russell and Norvig, 2003] Russell, S. and Norvig, P. (2003). *Artificial Intelligence. A Modern Approach*. Prentice Hall.

[Schwall and Gerdes, 2002] Schwall, M. and Gerdes, C. (2002). A probabilistic Approach to Residual Processing for Vehicle Fault Detection. In *Proceedings of the 2002 ACC*, pages 2552–2557.

[Sheppard and Kaufman, 2005] Sheppard, J. W. and Kaufman, M. A. (2005). A Bayesian Approach to Diagnosis and Prognosis Using Built-In Test. *IEEE Transactions on Instrumentation and Measurement*, 54:1003–1018.

[Silander and Myllymäki, 2006] Silander, T. and Myllymäki, P. (2006). A Simple Approach for Finding the Globally Optimal Bayesian Network Structure. In *proceedings of UAI*.

[Svärd and Nyberg, 2009] Svärd, C. and Nyberg, M. (2009). Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems. To appear in *IEEE Transactions on Systems, Man and Cybernetics. Part A*.

[Weber et al., 2006] Weber, P., Theilliol, D., Aubrun, C., and Evsukoff, A. (2006). Increasing Effectiveness of Modelbased Fault Diagnosis: a Dynamic

Bayesian Network Design for Decision Making. In *Proceedings of 6th IFAC Symposium on Fault Detection, Supervision and Safety of technical processes (SAFEPROCESS 2006)*, pages 109–114.

Part IV

Concluding Remarks

5

Concluding Remarks

It is a capital mistake to theorize before you have all the evidence. It biases the judgment.

Sherlock Holmes, 1888. In "A Study in Scarlet"

1 Conclusions

The main objective with this thesis has been to contribute to improved diagnosis of automotive vehicles. The work been driven by case studies of real applications, such as automotive engines and breaking systems. We have studied both on-board diagnosis, in Paper 1, 2, and 5, and off-board diagnosis for troubleshooting, in Paper 3 and 4. In the case studies, challenges and problems have been identified. In both on- and off-board diagnosis, the limited amount of training data and the uncertainties in models of the system are two of the most important challenges. To face these challenges we have chosen a probabilistic approach, and compute the *probabilities* that faults are present in the system under diagnosis.

Considering on-board diagnosis, two of the most important issues are to handle the experimental training data, need for integration of different kinds of knowledge of the diagnosed system, and the hardware capacity limitations. In Paper 1 a method for combining expert knowledge in terms of a Fault Signature Matrix (FSM) with experimental training data has been developed, and in Paper 2 a method that combine likelihood constraints with data has been

developed. Both these methods are generic, and applicable to several different fields of applications.

In Paper 5 five approaches resulting in eight methods for learning fault diagnosis and isolation has been compared. The comparison is made with on-board diagnosis in mind, but they are applicable also to off-board diagnosis. In the survey, the methods based on logistic regression have proved to have the best performance, in particular in relation to the small number of parameters needed.

In off-board diagnosis for troubleshooting, we have identified three main issues: in models for troubleshooting there are both instant and non-instant edges, the need for computing probabilities of variables in a system that is subject to interventions, and the need for time efficient probability computations. This has led to the development of the framework of event-driven non-stationary dynamic Bayesian networks (nsDBN) in Paper 3, and its further development in Paper 4 to the algorithm *updateBN* that is optimized for probability computations in troubleshooting. The framework of event-driven nsDBNs is a general framework for modeling processes with external interventions, and is applicable not only to troubleshooting.

In Chapter 1 we formulated the problem to be solved in the thesis as five questions. We are now ready to answer these.

- *How do standard methods for learning from data perform in the computation of* (1.1)*?* For eight methods from five different approaches, including different types of Bayesian networks and regression, this question is answered in Paper 5. Of course, the results are dependent on the particular diagnosis situation, but one conclusion can be drawn: it is important that the method handles experimental data. Furthermore, methods with a smaller number of parameters perform better than those with more parameters.

- *Which are the main issues regarding the training data available for diagnosis?* Training data is used in Paper 1, 2, and 5, and in these papers we have identified two main challenges: (a) the lack of data from faults and fault combinations that are to be diagnosed, and (b) the fact that data is experimental. The handle (a), methods for combining data and knowledge are crucial. In particular, in Paper 1 and 2 experiments have shown that combining both data and expert knowledge improves the inference, compared to using data alone. The fact (b), that data is experimental, means that no information about the prior distribution of faults (before observations are made) can be learned from the data. In all Paper 1, 2, and 5 the experimental training data is handled in different ways, depending on the over-all strategy in each of the papers. However, all three methods allows for integration of prior probabilities.

- *How should these different kinds of information be integrated in the computations?* In Paper 1 and 2, it has been shown that two of the most common types of expert knowledge in diagnosis, namely in form of an FSM and in form of relations between conditional distributions of single observations, appears as different kinds of constraints in the computations. In particular, the relations between single observations are translated to likelihood constraints, and it is shown that the likelihood constraints can be used to represent a broad class of information

- *What are the effects of the different kinds of dependencies?* In probability computations for troubleshooting with interventions this has been shown to be a very important question to get the probabilities right. In Paper 3 and 4, the concepts of instant and non-instant dependencies and persistent and non-persistent variables are introduced to handle this task.

- *How should external interventions be handled in the computation of* (1.1)*?* In Paper 3 and 4 the event-driven non-stationary nsDBNs and the algorithm *updateBN* been developed as an answer to this question.

- *How to compute the probability* (1.1) *as efficiently as possible?* This question is very difficult, or even impossible, to answer, since it depends on the available information and the requirements on the accuracy of the computed result. However, in all five papers in the thesis it has been one main issue to limit computation time and, in particular when considering on-board diagnosis, to optimize storage requirements.

2 Future Research

In this thesis, steps have been taken towards the use of probabilistic methods for diagnosis in automotive applications. Although answering several questions, including the five listed in Section 1, many new have appeared during the work. In this section we make an outlook on future work and research using a broad and holistic view. Detailed suggestions on future work are presented in each of the five appended papers.

Other Background Knowledge. In the thesis, we have considered background knowledge in terms of a Fault Signature Matrix in Paper 1, and in terms of likelihood constraints in Paper 2. These two types of background knowledge are general and can describe many types of expert knowledge. It is shown in the papers that the same kind of background knowledge appears in many different areas of applications. A natural next step is to investigate which other kinds of background knowledge that exist, and how they can be combined with data in probability computations. Furthermore, to increase the possibility

of diagnosing and isolating faults, it would be interesting to combine different kinds of background knowledge with each other.

Finding Dependencies and Numbers. In probabilistic models, both the structure of dependencies between variables and the underlying conditional probability distributions need to be determined. Data could be used to learn the models, but as discussed in the thesis, the amount of data is often limited. In particular this is true when systems are under development or freshly released to the market. In addition, engineers that develop an automotive system possess a large amount of knowledge and intuition about it. To use their knowledge in the diagnosis, it must be translated to a form that can be used in the probability computations. To get the most out of the probability computations, future research concerning the translation of experts' knowledge to probability distributions is interesting. This is particularly important in modeling for troubleshooting as in Paper 3 and 4.

Fault Tolerant Control. In Paper 4 we have discussed troubleshooting from a decision-theoretic view-point, and combined probabilities for faults with loss functions to compute the best action for a workshop mechanic to perform. Similarly, for on-board diagnosis, it would be interesting future work to combine probability computations with loss-functions. Interesting future work would be to apply this approach also in Fault Tolerant Control (FTC), where the objective is to control the control-systems in the vehicle to avoid damaging consequences of faults.

Performance Measures. In order to compare and evaluate diagnosis methods, performance measures are necessary. In the literature there are several performance measures, such as percentage of correct classification, log-loss-scoring function, or mean-square error, see for example [Devroye et al., 1996, Gustafsson, 2001]. However, these are general performance measures, and not developed for diagnosis. Is it the case that a fault diagnosis system with good score in these performance measures performs well in diagnosis? Furthermore, what is a desired behavior of a fault diagnosis algorithm? The answer depends, of course, on how the output from the diagnosis system is supposed to be used. Indeed, the probability for a fault itself is rather uninteresting, as long as no reaction on the fault is suggested or performed. Therefore, one attractive alternative is to combine the fault diagnosis algorithm with a loss function and compute the expected loss, or the risk. For example, in troubleshooting the Expected Cost of Repair, defined in Paper 4 could be a suitable performance measure. Future work in this area includes for example finding proper loss functions and evaluating diagnosis systems to understand which properties that gives high scores.

Scaling. In all five appended papers, as in many research areas, the problems related to scaling of the methods to larger problems are identified as important and interesting future work. A related question is whether the method can be applied to subsystems and the results from each subsystem combined, instead of scaling the methods to larger systems?

Other Data Driven Methods and Training Data. In Papers 1, 2, and 5 we have focused on learning from data, and focused on probabilistic methods in general and Bayesian methods in particular. In the literature, there are also other methods for retrieving knowledge from data, such as Support Vector Machines, Neural Networks, and Nearest Neighbor-methods, see for example [Duda et al., 2001, Bishop, 2005]. Work has been performed on applying such methods to the diagnosis task, see for example [Russell et al., 2000, Verron et al., 2007, Lee et al., 2007]. However, these methods are based on data only, and expert knowledge of the kinds used in Papers 1 and 2 are not used. Interesting future work include applying these methods to fault diagnosis, and investigate how expert knowledge can be integrated in these methods.

References

[Bishop, 2005] Bishop, C. M. (2005). *Neural Networks*. Oxford University Press.

[Devroye et al., 1996] Devroye, L., Györfi, L., and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer, New York.

[Duda et al., 2001] Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley, New York.

[Gustafsson, 2001] Gustafsson, F. (2001). *Adaptive Filtering and Change Detection*. Wiley.

[Lee et al., 2007] Lee, G., Bahri, P., Shastri, S., and Zaknich, A. (2007). A Multi-Category Decision Support System Framework for the Tennessee Eastman Problem. In *Proceedings of the European Control Conference (ECC 07)*.

[Russell et al., 2000] Russell, E. L., Chiang, L. H., and Braatz, R. D. (2000). *Data-Driven Techniques for Fault Detection and Diagnosis in Chemical Processes*. Springer.

[Verron et al., 2007] Verron, S., Tiplica, T., and Kobi, A. (2007). Fault Diagnosis of Industrial Systems with Bayesian Networks and Mutual Information. In *Proceedings of the European Control Conference (ECC 07)*, pages 2304–2311.

A

Interpretations of Probability

Life's most important questions are, for the most part, nothing but probability problems

Laplace, 1814

Computations with probabilities follow well defined rules, such as the Sum Rule, the Product Rule, and Bayes' Rule [Blom, 1994, Durrett, 2004, Casella and Berger, 2001]. However, to use these tools for computing probabilities, it is necessary to find the numbers on conditional probabilities and prior probabilities. To determine these numbers, it is necessary to know what the "probability" really is.

A.1 Dealing With Uncertainty

Human life is to a great deal a life lived under uncertainty. Every day we make decisions under uncertainty, both in professional life and in private. For example: will the stock market raise or fall today? My car does not start, which part has caused the failure? Should I bring an umbrella tonight? How much should I bet on my favorite soccer team in the next game? Should I fold in the poker game? What conclusions can be drawn from the laboratory experiment? There is no upper limit on the number of such situations.

The situations listed above are very different in their nature. Sometimes the probability calculation relies on data, as in laboratory experiments. In other cases the probability calculations are based on known facts, for example, the

number of spades in a deck of card is well known and thus the probability of drawing a spade can be computed. In yet other cases, it seems like probabilities are more or less based on personal feelings, for example in sports betting.

In each situation, the human brain deals with uncertainty. It considers the available information, for example: yesterday's stock market trend or the observation that the headlights of my car does not light. The brain weighs factors speaking fore and against an event, and makes decisions (which may be more or less clever).

In the problem considered in this thesis, diagnosis of automotive vehicles, we deal with uncertainty in a formal way. Given observations of different kinds from a system, the aim is to construct an algorithm that, just like the human brain, considers the available information and evaluates the probabilities that different faults are present. The available information can for example comprise data, different kinds of models with unknown model errors, drawings, and functionality specification documents. To be able to transform these fundamentally different types of information and construct the diagnosis algorithm that computes probabilities for faults, one might ask oneself questions as: What is this "uncertainty"? What is "probability"? What does the "probability that it will rain tonight" mean? Is it unique? Can we put a number on it? In reference literature on probability theory, for example [Blom, 1994, Durrett, 2004, Casella and Berger, 2001, O'Hagan and Forster, 2004], formulas and tools for manipulating probabilities are presented, as in the following toy example.

Example A.1.1 (Was it the Sprinkler?).
Sanna wakes up a morning and wants to know whether it has rained during the night. She knows that the prior probability for rain is $p(rain) = 0.3$. Moreover, she knows that, if it has rained, the lawn will be wet, i.e. that $p(wet\ lawn|rain) = 1$. She also knows that, if there is no rain, there is a sprinkler that cause the lawn to be wet with probability $p(wet\ lawn|no\ rain) = 0.2$.
After waking up, Sanna notices that the lawn is wet. She can then compute the probability that it has rained by using Bayes' rule and marginalization [Blom, 1994] as follows:

$$\begin{aligned} p(rain|wet\ lawn) &= \frac{p(wet\ lawn|rain)p(rain)}{p(wet\ lawn)} = \\ &= \frac{p(wet\ lawn|rain)p(rain)}{p(wet\ lawn|rain)p(rain) + p(wet\ lawn|no\ rain)p(no\ rain)} = \\ &= \frac{1 \cdot 0.3}{1 \cdot 0.3 + 0.2 \cdot 0.7} = 0.6818\ldots \end{aligned}$$

These computations are perfectly fine as long as the numbers, such as "the probability for rain is 0.3", are known. In the example above, the numbers where simply stated, but how are they found? To assign numbers in the probability

distributions to use in computations, it is necessary to know what "probability" means.

A.2 Interpretations of Probability

The discussion about the definition of the word "probability" has been going on for more than 200 years [Hacking, 1976]. Depending on the background of the researchers, there were several different interpretations during these years. The first rigorous description of probability is often considered as the one given by Pierre-Simon Laplace [Laplace, 1951] in 1814:

> *The theory of chance consists in reducing all the events of the same kind to a certain number of cases equally possible, that is to say, to such as we may be equally undecided about in regard to their existence, and in determining the number of cases favorable to the event whose probability is sought. The ratio of this number to that of all the cases possible is the measure of this probability, which is thus simply a fraction whose numerator is the number of favorable cases and whose denominator is the number of all the cases possible.*

Since Laplace's definition the reactions and discussion about the meaning of the word "probability" has been numerous. No consistent definition of the word exists, instead *interpretations* are considered. The clash of opinions was commented by Savage [Savage, 1954] in 1954:

> *As to what probability is and how it is connected with statistics, there has seldom been such complete disagreement and breakdown of communication since the Tower of Babel.*

A.2.1 Bayesians and Frequentists

The discussion about the definition of the word "probability" has been going on for more than 200 years [Hacking, 1976]. Depending on the background of the researchers, there where several different interpretations during these years. Among the different interpretations of probability, there are two main paths [Hacking, 1976, O'Hagan and Forster, 2004, Jaynes, 2001]: the idea of probability as a *frequency* in an ensemble, often called the *frequentist* view or *frequency-type*, on the one hand; and the idea of probability as the *degree of belief* in a proposition, often referred to as the *Bayesian* view or *belief-type*, on the other hand. There are several labels on the two interpretations of probability, such as subjective/objective, epistemic/aleatory, belief-type/frequence-type, Number 1/Number 2 [Hacking, 1976].

In frequentist view, probability is defined by the relative frequency of an event, and is a property of the object. Consider for example the statement:

> *This coin is biased towards heads. The probability of getting heads is about 0.6.*

This statement expresses probability in the frequency-type meaning, and is true depending on "how the world is". This statement can (at least hypothetically) be tested by tossing the coin (infinitely) many times. If the relative frequency for heads is 0.6, the statement is true, if the relative frequency for heads is something else, the statement is false. In the Bayesian view, probability is the degree of belief, given some evidence. Consider now this sentence about the same coin:

> *Taking all the evidence into consideration, the probability of getting a head in the next roll is about 0.6.*

This statement is true depending on how well evidence supports the particular probability assignment. The probability is subjective in the sense that it depends on the evidence. This statement can be true, depending on the evidence, even if the relative frequency turns out to be something else than 0.6.

For a dogmatic frequentist, probabilities exists only when dealing with experiments that are random and well-defined. The probability of random event is defined as the relative frequency of occurrence of the outcome of the experiment, when repeating the experiment infinitely many times [Hacking, 1976]. Famous frequentists are Jerzy Neyman, Egon Pearson, and Ronald Aylmer Fisher.

In the frequentist interpretation, the probability of an event is a property of the event, and it is well defined only for events that can be repeated infinitely many times. Thus, questions such as "what is the probability for rain tomorrow?" are not defined, because there is only one today and one tomorrow, and it is impossible to construct repeated experiments to investigate the relative frequency of rainy days the day after today[1]. However, asking the weather office the answer would be something like "It is mid december, and it was rain yesterday. During the last thirty years there has been rain 50% of the days, and of those days, there has been rain the following day for about 50%". Thus, in a frequentistic view, the probability of rain tomorrow is the probability of rain a "general day in mid December, where it has been rain the day before", rather than *tomorrow*. This is a different interpretation from the Bayesian view.

In the Bayesian view, probabilities can be assigned to any statement, regardless of whether there is any random process involved. The probability of an

[1] In his book [Jaynes, 2001], Jaynes takes this argument even further and claims that there are (almost) no experiments that can be controlled so perfectly that it is guaranteed that they are repetitions of the same event.

event represents an individuals degree of belief in that event, given all information that the individual has at hand. In the Bayesian view, the probability is a property of the spectator and in particular the information the spectator has at hand, and not a property of the event. Famous Bayesians are for example Bruno de Finetti, Frank Ramsey, L. J. Savage, and Edwin T. Jaynes. The difference between the frequentist and Bayesian view is illustrated in the following example.

Example A.2.2 (Urn Experiment - Frequentists vs. Bayesians).
Statement S: "There is an urn with equally many white and black balls." For a frequentist $F1$ the probability of drawing a white ball is 0.5, since if balls where drawn from the urn infinitely many times half of them would be white. For a Bayesian $B1$ with information S, the probability of drawing a white ball is 0.5, since there is no reason for the Bayesian to favor white or black[2]. For a Bayesian $B2$ with information S together with the statement S_1: "the black balls where put into the urn before the white balls", would have a higher probability for drawing a white ball than Bayesian $B1$.

The urn example above illustrates two important things. First, and proven in [O'Hagan and Forster, 2004], for repeatable and independent random events, such as drawing a ball from an urns, the Bayesian and frequentist views coincidence. Also, all computational rules of probabilities, such as the product rule, the sum rule, and Bayes' rule can be used with both frequentist and Bayesian definitions of probabilities.

Second, it is clear that for Bayesian $B2$ with information S_1 in addition to S has another probability for drawing a white ball than Bayesian $B1$ has. However, the exact value of the probability for Bayesian $B2$ is not easily determined. For a frequentist, the probability of an event is defined as its relative frequency. It is a property of the object, and is sometimes said to be objective. For a Bayesian the probability for an event is *subjective* in the sense that it is determined by the information the person has at hand. However, as discussed in the next section, the probability for an event is not *arbitrary*.

A.2.2 Switching Between Interpretations

These two views, the frequency-type and the (Bayesian) belief-type, are different in a philosophical sense, and a natural question is why the same word, "probability", is used for both of them. Hacking [Hacking, 1976] gives one explanation: in daily life, we (humans) switch back and forth between the two perspectives. Consider the following example.

Example A.2.3 (Switching Between Frequency and Belief).
A truck of model R arrives to a mechanic at a workshop. The mechanic knows

[2]This is often called the *Principle of Indifference*.

that among all model R trucks, one out of ten of the trucks that arrives to the workshop has fault F present. The mechanic concludes that choosing a random model R truck of those that has been (or are) at the workshop, the probability that fault F is found is 0.1. This probability is of frequency-type.
Consider now the particular truck that just arrived to the workshop. What is the probability that *this* truck is has fault F? The truck *is* either faulty of fault free, so there is no randomness, but still the mechanic would (probably) say that the probability is 0.1. He reasons as follwos. Out of all model R trucks that has visited the workshop, fault F was present in 1 out of 10. This truck is a model R and has arrived to the workshop. Taking those three pieces of information into account, the probability that this particular truck has fault F is 0.1.

A.3 The Bayesian View: Probability as an Extension to Logic

In this section, we follow the reasoning by Jaynes in [Jaynes, 2001], and show how the belief-type, (or Bayesian) interpretation of probability can be subjective without beeing arbitrary. To do this, we use the language of logic, and extend it to also consider uncertain events. For example, assume that it is known that $A \Rightarrow B$, and that we know that the event A is true. We can then draw the conclusion that also B is true. On the other hand, if B is known to be true we can not say anything about A with certainty. However, our common sense says that if B is known to be true, A is *more likely* to be true.

A.3.1 Consistency and Common Sense

We will now formalize this reasoning, but first we recall the traditional definition of probability. by Kolmogorovs axioms [Blom, 1994, Jaynes, 2001]:

- For every event A it holds that $p(A) \in [0, 1]$.
- For the whole sample space Ω it holds that $p(\Omega) = 1$.
- If A and B are mutually exclusive, it holds that $p(A \cup B) = p(A) + p(B)$ ("Sum Rule").

Furthermore, the conditional probability of A given B is *defined* by

$$p(A|B) = \frac{p(AB)}{p(B)} \text{ "Product Rule".}$$

Jaynes [Jaynes, 2001], based on Cox [Cox, 1946], takes another approach. Starting from three fundamental desiderata, including requirements on consistency and common sense, they show that probability *must* fulfill the sum and product rules. The three desiderata are:

I Degrees of plausibility are represented by real numbers.

II Qualitative agreement with common sense.

III Consistency:

> **(a)** If a conclusion can be reasoned out in more than one way, then every possible way must lead to the same result.
>
> **(b)** All evidence relevant to the question should be taken into account. Some of the information can not be arbitrary ignored and the conclusions drawn on what remains.
>
> **(c)** Equivalent states of knowledges should always be represented by equivalent plausibility assignments. That is, if in two problems the state of knowledge is the same, then the same plausibilities must be assigned in both.

In the desiderata above, uncertainty is expressed in terms of *plausibility*. In [Jaynes, 2001] Jaynes states that probability is a monotonic function p of plausibility. Adding the requirement that probability should be described by a real number between 0 and 1, and adopting the convention that 1 represents that an event is true with certainty, and 0 that an event is certainly false, Jaynes [Jaynes, 2001] shows how the rules for probability computations can be computed from the three desiderata given above. In particular, this holds for the sum rule and the product rule.

The results in [Jaynes, 2001] and [Cox, 1946] are criticized and debated for example in [Halpern, 1999] and [Arnborg and Sjödin, 2000]. However, as remarked by Arnborg and Sjödin in [Arnborg and Sjödin, 2000], the "authors advocating standard Bayesianism have not been strengthened or weakened" by their analysis.

A.3.2 The Statements Behind the |-sign

In the Bayesian view, the probability of an event is determined uniquely by the information behind the |-sign. In [Jaynes, 2001], Jaynes argues that it is nonsense to talk about the probability of an event A without expressing the information $\mathbf{i}$ which it is based on. Even if there are no other explicit

events available, $\mathbf{i}$ includes general information, for example about how prior probabilities are assigned.

A.4 Assigning Numbers

In the discussion above we have seen that there are two main interpretations of probabilities, the frequentist and the Bayesian view, and as discussed in Chapter 3 we switch between these interpretations in ever-day life. We have discussed that also when the relative frequency, i.e. the probability according to the frequentist view, is not defined or relevant, we can use the Bayesian, belief-type, view. However, in the Bayesian case, there is still one main challenge left: How to assign numbers to the probabilities?

In order to obtain a non-arbitrary theory for probabilities, we need objective ways for determining the numbers. There are two cases to consider: (i) assigning probability distribution for a variable X, given a certain state of knowledge $\mathbf{i}^*$; and (b) assigning probabilities of an event A given a certain state of knowledge $\mathbf{i}^*$.

The "state of knowledge" may be a defined background knowledge, for example "I rolled this dice yesterday, and it showed five eyes" if A is the event "roll the dice and obtain six eyes". However, $\mathbf{i}^*$ often represent the "prior knowledge" about A (or X). In many situations, the prior knowledge is used to express ignorance, i.e. "knowing nothing". In this case, the prior probability distribution, the probability distribution conditioned on $\mathbf{i}^*$ only, should be *non-informative*.

In the following sections we present four commonly used approaches for assigning prior probability distributions, followed by a method for assigning probabilities. Methods for assigning priors is further discussed for example in [O'Hagan and Forster, 2004].

A.4.1 Principle of Indifference

Suppose that there are $n > 1$ possible events, the principle of indifference then says that if there is no reason for favoring any of the events over the others, each event should be assigned probability $1/n$ (see [Jaynes, 2001, O'Hagan and Forster, 2004]). The Principle of Indifference is sometimes called the *Principle of Insufficient Reason*.

A.4.2 Jeffreys Prior

Jeffreys Prior for a real-valued variable X is given by $p(x) = 1/x$. It is an improper prior, i.e. it does not integrate to one. However, since prior probability distributions are always used together with likelihoods $p(x|y)$ to obtain a posterior probability $p(y|x) \propto p(x|y)p(x)$, they can be safely used [O'Hagan and Forster, 2004].

Jeffreys prior has two interesting properties. First, it is invariant to scaling of x, and, second, it is uniform in the logarithm of x. With Jeffreys prior, the probability of obtaining a number in the interval $[1, 10]$ is equal to the probability of obtaining a number in the interval $[10, 100]$.

A.4.3 Maximum Entropy

A more general approach to assigning prior probabilities, is to use the concept of entropy [Jaynes, 2001],

$$H_p(x) = -\sum_i p(x_i|\mathbf{i}^*) \log p(x_i|\mathbf{i}^*), \tag{1}$$

where the sum is replaced by an integral sign in the continuous case. The idea is to use the distribution p^* that is consistent with the available information $\mathbf{i}^*$ and that maximizes H_p.

A.4.4 Reference Priors

A method, related to the maximum entropy method, for assigning priors is the concept of reference priors introduced by Bernardo in [Bernardo, 1979]. Bernardo considers the problem of probability updating, i.e the computation of the probability of x after learning y, given by

$$p(x|y, \mathbf{i}^*) = \propto p(y|x, \mathbf{i}^*)p(x|\mathbf{i}^*). \tag{2}$$

The likelihood $p(y|x, \mathbf{i}^*)$ in the equation above is assumed to be known, and $p(x|\mathbf{i}^*)$ is the prior to be assigned.

The reference prior is the "least informative" prior in the sense that as much as possible is learned about X through the likelihood. This means that the difference in information (or knowledge) about X in the posterior distribution $p(x|y, \mathbf{i}^*)$ relative to the prior $p(x|\mathbf{i}^*)$ is maximized. The reference prior is obtained by maximizing the expected Kullback-Leibler divergence of the posterior distribution relative to the prior. Technically, the reference prior is defined in the asymptotic limit, i.e., the limit of the priors obtained by maximizing the expected Kullback-Leibler divergence to the posterior as the number of data points goes to infinity.

A.4.5 Betting Game

On possibility for assigning probabilities to events that are not possible to repeat several times is to use a betting exercise as described in [Jensen and Nielsen, 2007, Jeffrey, 2004]. For example, what is the probability that there will be snow in Linköping on December 18 2010? Anna, based on her background and

experience, estimate the probability to p_A. Bill, with other background knowledge and other experience, may estimate the probability to another value, say p_B. In this sense, the probability for snow in Linköping 2010 is *subjective*. One way to assign numbers to the subjective probabilities is the following. Assume that there is a ticket that is worth €100 each if there is snow in Linköping on December 2010. Anna thinks that €10 is the right price for this ticket, and thus $p_A = 0.1$. Bill, on the other hand, may think that €1 is just the right price, and thus $p_B = 0.01$. Betting games as the one described above are used to predict markets for commercial means [Hanson, 2007].

References

[Arnborg and Sjödin, 2000] Arnborg, S. and Sjödin, G. (2000). On the Foundations of Bayesianism. In *Bayesian Inference and Maximum Entropy Methods in Science and Engineering 20th International Workshop*.

[Bernardo, 1979] Bernardo, J. (1979). "reference posterior distributions for bayesian inference". *Journal of the Royal Statistical Society*, 41(2):113–147.

[Blom, 1994] Blom, G. (1994). *Sannolikhetsteori och statistik med tillämpningar*. Studentlitteratur.

[Casella and Berger, 2001] Casella and Berger (2001). *Statistical Inference (2nd edition)*. Duxbury Press.

[Cox, 1946] Cox, R. T. (1946). Probability, Frequency, and Reasonable Expectation. *American Journal of Physics*, 14:1 – 13.

[Durrett, 2004] Durrett, R. (2004). *Probability: Theory and Examples*. Duxbury Press.

[Hacking, 1976] Hacking, I. (1976). *The Logic of Statistical Inference*. Cambridge University Press.

[Halpern, 1999] Halpern, J. Y. (1999). A Counterexample to Theorems of Cox and Fine. *Journal of Artificial Intelligence Research*, 10:67–85.

[Hanson, 2007] Hanson, R. (2007). Logarithmic market scoring rules for modular combinatorial information aggregation. *The Journal of Prediction Markets*, 1(1):3–15.

[Jaynes, 2001] Jaynes, E. T. (2001). *Probability Theory - the Logic of Science*. Camebridge University Press, Cambridge.

[Jeffrey, 2004] Jeffrey, R. C. (2004). *Subjective Probability: the Real Thing*. Cambridge.

[Jensen and Nielsen, 2007] Jensen, F. V. and Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs*. Springer.

[Laplace, 1951] Laplace, P.-S. (1814 (English edition in 1951)). *A Philosophical Essay on Probabilities*. Dover Publications Inc, New York.

[O'Hagan and Forster, 2004] O'Hagan, A. and Forster, J. (2004). *Kendall's Advanced Theory of Statistics*. Arnold, London.

[Savage, 1954] Savage, L. J. (1954). *The foundations of Statistics*. John Wiley & Sons, Inc., New York.

Index

Linköping Studies in Science and Technology
Department of Electrical Engineering
Dissertations, Vehicular Systems

No. 12 *A DAE formulation for Multi-Zone Thermodynamic Models and its Application to CVCP Engines*
Per Öberg, Dissertation No. 1257, 2009.

No. 11 *Control of EGR and VGT for Emission Control and Pumping Work Minimization in Diesel Engines*
Johan Wahlström, Dissertation No. 1256, 2009.

No. 10 *Efficient Simulation and Optimal Control for Vehicle Propulsion*
Anders Fröberg, Dissertation No. 1180, 2008.

No. 9 *Single-Zone Cylinder Pressure Modeling and Estimation for Heat Release Analysis of SI Engines*
Markus Klein, Dissertation No. 1124, 2007.

No. 8 *Modeling for Fuel Optimal Control of a Variable Compression Engine*
Ylva Nilsson, Dissertation No. 1119, 2007.

No. 7 *Fault Isolation in Distributed Embedded Systems*
Jonas Biteus, Dissertation No. 1074, 2007.

No. 6 *Design and Analysis of Diagnosis Systems using Structural Methods*
Mattias Krysander, Dissertation No. 1033, 2006.

No. 5 *Air Charge Estimation in Turbocharged Spark Ignition Engines*
Per Andersson, Dissertation No. 989, 2005.

No. 4 *Residual Generation for Fault Diagnosis*
Erik Frisk, Dissertation No. 716, 2001.

No. 3 *Model Based Diagnosis: Methods, Theory, and Automotive Engine Applications*
Mattias Nyberg, Dissertation No. 591, 1999.

No. 2 *Spark Advance Modeling and Control*
Lars Eriksson, Dissertation No. 580, 1999.

No. 1 *Driveline Modeling and Control*
Magnus Pettersson, Dissertation No. 484, 1997.