

A Bayesian Approach to Learning in Fault Isolation

Anna Pernestål **Hannes Wettig, Tomi Silander** **Mattias Nyberg** **Petri Myllymäki**
Dept. Electrical Engineering Helsinki Institute for Dept. Electrical Engineering Helsinki Institute for
Linköping University Information Technology Linköping University Information Technology
Sweden Finland Sweden Finland
annap@isy.liu.se *{wettig,tsilande}@hiit.fi* *matny@isy.liu.se* *petri@hiit.fi*

Abstract

Fault isolation is the task of localizing faults in a process, given observations from it. To do this, a model describing the relation 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, as well as 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; the diagnosis of an automotive engine.

1 Introduction

We consider fault isolation, i.e. the task of localizing faults that are present in a process given observations from this 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 physical knowledge about the process, and represented as a structure 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 also data available from the process. In the current work we investigate and compare different methods for learning the model for diagnosis from training data, and the possibilities to integrate them with the expert knowledge.

We are motivated by the problem of fault isolation in an automotive engine, and we have used a Scania diesel engine as source for training and evaluation data. In engine fault isolation there may be several hundreds of faults and observations. There will be 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 also 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. 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 of the current work 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 on 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. Consecutively we show how the five methods can be adopted to the isolation problem. We apply them to the task of fault isolation in an automotive diesel engine.

Bayesian methods for fault isolation have been previously studied in literature. In these previous works it is generally 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 typically rely on pattern recognition methods described for example in [Bishop, 1995; 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]. These methods are applicable if there is sufficient 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 missing fault patterns in the data. In the current paper we take a Bayesian approach to learning

for fault isolation, which 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. In [Pernestål and Nyberg, 2007] training data is combined with fundamental methods for fault isolation described in [de Kleer and Williams, 1992; Reiter, 1992]. This approach 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 prediction 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 $\mathbf{Y} = (Y_1, \dots, Y_K)$, where $Y_k = 1$ means that fault k is present, and let the observations be represented by the variables $\mathbf{X} = (X_1, \dots, X_L)$, 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. Bold-face letters denote vectors.

We write $p(\mathbf{X} = \mathbf{x})$ (or simply $p(\mathbf{x})$) to denote either probabilities, 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. A Bayesian network encodes the joint probability distribution over a finite set of variables $\{X_1, \dots, X_L\}$, and decomposes it into a sequence of conditional probability distributions, one for each variable.

More specifically, let $\text{pa}(X_i)$ denote the parents of X_i , and let $\text{pa}(x_i)$ be a value (configuration) of $\text{pa}(X_i)$. Then there is a conditional probability distribution $p(x_i|\text{pa}(x_i))$ for each variable X_i . Nodes without parents are called *root nodes*, and their conditional probabilities are simply their prior probability $p(x_i)$. The joint probability distribution of $\{X_1, \dots, X_L\}$ can be obtained by taking the product of all these conditional probability distributions:

$$p(x_1, \dots, x_L) = \prod_{i=1}^L p(x_i|\text{pa}(x_i)). \quad (1)$$

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 edges in the Bayesian network, it also includes all the distributions $p(x_i|\text{pa}(x_i))$. When we discuss learning in Bayesian networks, we mean learning both the structure and the probability distributions.

3 Bayesian Fault Isolation

We are now ready to state the fault isolation problem in probabilistic terms, and present relevant performance measures.

3.1 Problem Formulation

Except the current observation \mathbf{X} from the process, we are also given a set of training data \mathcal{D} . Training data consists of samples $(\mathbf{y}^n, \mathbf{x}^n)$, $n = 1, \dots, N_{\mathcal{D}}$, of pairs of fault and observation variables. The training data is collected by implementing faults and then collecting observations, meaning that training data is *experimental*. To evaluate the fault isolation methods we use a set \mathcal{E} consisting of $N_{\mathcal{E}}$ samples. The evaluation data is collected by running the process without integrating with it (i.e. without implementing any faults but rather observing faults as they appear), meaning that evaluation data is *observational*. Furthermore, 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

$$p(\mathbf{y}) = \prod_{k=1}^K p(y_k|y_1, \dots, y_{k-1}) \approx \prod_{k=1}^M p(y_k), \quad (2)$$

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 factors, such as humidity, driver behavior, and operation point of the process. These unobserved factors could be modeled using hidden nodes, but since they are numerous and unknown they are here approximated with dependencies between observation variables. This is more carefully discussed in [Pernestål *et al.*, 2006].

In the current work we take a Bayesian view point 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 I , i.e. to compute the probabilities $p(y_k|\mathbf{x}, \mathcal{D}, I)$, $k = 1, \dots, K$. The probability for a

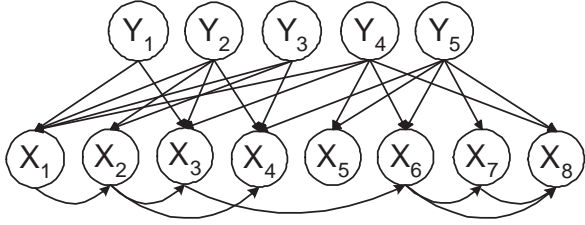


Figure 1: A Bayesian network describing a typical fault isolation problem.

fault y_k can be found by marginalizing over all other faults $\mathbf{y}_{-k} = (y_1, \dots, y_{k-1}, y_{k+1}, \dots, y_K)$,

$$p(y_k|\mathbf{x}, \mathcal{D}, I) = \sum_{\mathbf{y}_{-k}} p(\mathbf{y}_{-k}, y_k|\mathbf{x}, \mathcal{D}, I). \quad (3)$$

Note that $(\mathbf{y}_{-k}, y_k) = \mathbf{y}$, and (3) means that we seek the conditional distribution $p(\mathbf{y}|\mathbf{x}, \mathcal{D}, I)$. To simplify the notation we will not write out the prior knowledge I explicitly in the equations.

Computing the conditional distribution $p(\mathbf{y}|\mathbf{x}, \mathcal{D})$ is generally difficult. Instead, we approximate it using a model \mathcal{M} , for example a Bayesian network or a regression model. For each model we need method for determining the parameters of the model. This means that we compute probabilities

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}, \mathcal{M}) = p_{\mathcal{M}(\mathcal{D})}(\mathbf{y}|\mathbf{x}), \quad (4)$$

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.

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, 1995; Mitchell, 1997]. The logistic score based on a set \mathcal{E} of evaluation data it is given by

$$\mu(\mathcal{E}, \mathcal{M}) = \frac{1}{N_{\mathcal{E}}} \sum_{n=1}^{N_{\mathcal{E}}} \log p_{\mathcal{M}(\mathcal{D})}(\mathbf{y}^n|\mathbf{x}^n). \quad (5)$$

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. In the fault isolation problem the conditional probabilities for faults are often combined with decision theoretic methods for troubleshooting [Heckerman *et al.*, 1995a], where optimal decision making requires conditional probabilities close to the generating distribution.

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, 1995]. We define

$$\nu(\mathcal{E}, \mathcal{M}) = \frac{|\mathcal{C}|}{N_{\mathcal{E}}}, \quad (6)$$

where $\mathcal{C} = \{n : y_k^n = 1, k = \arg \max_{k'} p_{\mathcal{M}(\mathcal{D})}(y_{k'}|\mathbf{x}^n)\}$,

and y_k^n denotes element k in \mathbf{y}^n . In words, \mathcal{C} is the set of all indices 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

All the methods considered in this paper – with the exception of DI – build separate models for each fault and thus assume independence among these. Before any training data is recorded the approximation corresponds to (2). Furthermore, since faults were inflicted in training data, the data does not include any information about co-occurrence of the faults. However, when we build separate models for each fault, we also make a stronger assumption, namely that the faults *remain* independent given the observations,

$$p(\mathbf{y}|\mathbf{x}) = \prod_{k=1}^K p(y_k|\mathbf{x}, y_1, \dots, y_{k-1}) \approx \prod_{k=1}^K p(y_k|\mathbf{x}) \quad (7)$$

This approximation is (after applying Bayes’ rule and canceling terms) equivalent to

$$p(\mathbf{x}|y_1, \dots, y_K) \approx \frac{1}{p(\mathbf{x})^{K-1}} \prod_{k=1}^K p(\mathbf{x}|y_k) \quad (8)$$

In (8) $p(\mathbf{x})$ 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, 2001]. 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 .

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}

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 of Section 5) that there is training data only from single faults. Using training data straight-forward 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 dependencies that irrelevant.

From Section 2 we know that it is assumed that there is at least one fault present. Let $\mathbf{Y} > 0$ denote that $\sum_k y_k > 0$, i.e. that there is at least one fault present. Similarly, let $\mathbf{Y} = 0$ denote $\sum_k y_k = 0$, i.e. that there is no fault present. The knowledge that there is at least one fault present recouples the single fault methods introduced in (7), since in general we have

$$p(\mathbf{y}|\mathbf{x}, \mathbf{Y} > 0) \neq \prod_k p(y_k|\mathbf{x}, \mathbf{Y} > 0), \quad (9)$$

To avoid this recoupling, we study the probability for the faults given the knowledge that at least one fault is present in detail. We have

$$p(\mathbf{y}|\mathbf{x}, \mathbf{Y} > 0, \mathcal{D}) = \frac{p(\mathbf{Y} > 0|\mathbf{y}, \mathbf{x}, \mathcal{D})p(\mathbf{y}|\mathbf{x}, \mathcal{D})}{p(\mathbf{Y} > 0|\mathbf{x}, \mathcal{D})} = \begin{cases} 0 & \mathbf{y} = 0, \\ \frac{p(\mathbf{y}|\mathbf{x}, \mathcal{D})}{1-p(\mathbf{Y}=0|\mathbf{x}, \mathcal{D})}, & \mathbf{y} = l \neq 0. \end{cases} \quad (10)$$

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 (10) in the evaluation phase.

4.2 Direct Inference

The first method for fault isolation that we present is the direct inference (DI). Similar to several previous fault isolation algorithms DI rely on prior knowledge about which observations 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 (k, l) means that fault Y_k can never affect observation X_l , while an \mathcal{X} mean that Y_k may affect observation X_l . DI aims at combining the information from the FSM with the training data

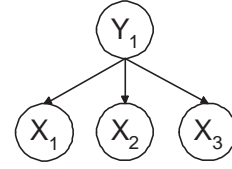


Figure 2: Naive Bayes network structure.

available. Assuming that observations are binary and that the background information I contains the FSM. Then, under certain assumptions it can be shown [Mitchell, 1997; Pernestål and Nyberg, 2007] that

$$p_{\text{DI}(\mathcal{D})}(\mathbf{y}|\mathbf{x}, \alpha_{\mathbf{x}\mathbf{y}}) = \begin{cases} 0 & \mathbf{x} \in \gamma \\ \frac{n_{\mathbf{x}\mathbf{y}} + \alpha_{\mathbf{x}\mathbf{y}}}{N_{\mathbf{y}} + A_{\mathbf{y}}} \frac{p(\mathbf{y}|I)}{\pi_0} & \text{otherwise,} \end{cases} \quad (11)$$

where π_0 is a normalization constant, $n_{\mathbf{x}\mathbf{y}}$ is the count in training data \mathcal{D} where the fault is \mathbf{y} and the observation is \mathbf{x} , $\alpha_{\mathbf{x}\mathbf{y}}$ 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 where true¹, $N_{\mathbf{y}} = \sum_{\mathbf{x}'} n_{\mathbf{x}'\mathbf{y}}$, and $A_{\mathbf{y}} = \sum_{\mathbf{x}'} \alpha_{\mathbf{x}'\mathbf{y}}$. The sets γ are determined by the background information as described in [Pernestål and Nyberg, 2007].

The DI method is 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 prediction, we model the joint distribution $p(\mathbf{y}, \mathbf{x}|\theta)$, where θ are parameters describing the conditional probability distributions in the network. From the joint distribution, the conditional distribution for each of the faults y_k can be computed. As described in Section 4, we build one model for each fault, combine them using (7) and correct for the knowledge that there is no fault present by using (10), and then we can marginalize to obtain the probability for each fault. We consider two types of Bayesian networks: Naive Bayes and general Bayesian Networks.

Naive Bayes

In a *Naive bayes* network it is assumed that the observations are independent given the fault. This structure is exemplified in Figure 2. 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 common methods used 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

¹the parameters α are sometimes referred to as *Dirichlet* parameters, since a Dirichlet prior is used in the computations

was first introduced in [Langley and Sage, 1994], where it is called *selective* naive bayes classifier. Let \mathcal{V} 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

$$V^* = \arg \max_{V \in \mathcal{V}} = \frac{1}{N_{\mathcal{D}}} \sum_{n=1}^{N_{\mathcal{D}}} \log p_{\mathcal{N}_V(\mathcal{D} \setminus \{y^n, \mathbf{x}^n\})}(y_k^n | \mathbf{x}^n, \alpha),$$

where α is the Dirichlet hyper-parameter for the NB model. 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, 1995], 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

$$p_{\mathcal{B}(\mathcal{D})}(y_k | \mathbf{x}, \alpha), \quad (12)$$

where α is the Dirichlet hyper-parameter for the BN model.

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}

$$p(\mathbf{y} | \mathbf{x}, \theta) = \frac{p(\mathbf{y}, \mathbf{x} | \theta)}{\sum_{\mathbf{y}} p(\mathbf{y}, \mathbf{x} | \theta)}. \quad (13)$$

It is well known [Ng and Jordan, 2002; Kontkanen *et al.*, 2001; Friedman *et al.*, 1997] that in such 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.

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,

$$y_k = \mathbf{w}_k^T \mathbf{x} + w_{k0} + \epsilon_k, \quad \epsilon \sim N(0, \sigma).$$

Here \mathbf{w}_k , w_{k0} , and σ are parameters to be determined. This gives the probability distribution

$$p_{\text{LinR}}(y_k | \mathbf{x}) = \frac{1}{Z} \exp\left(-\frac{(\mathbf{w}_k^T \mathbf{x} + w_{k0} - y_k)^2}{2\sigma^2}\right),$$

where Z is a normalization constant. To determine the parameters we use the standard methods described for example in [Bishop, 1995]. 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, 1995].

Logistic Regression

Learning parameters to maximize (13) 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 classifier². Let α and β be parameters in the logistic regression model, and define

$$p_{\text{LogR}}(Y_k = 1 | \mathbf{x}, \alpha, \beta) = \frac{\exp s(\mathbf{x}, \alpha, \beta)}{\exp s(\mathbf{x}, \alpha, \beta) + \exp -s(\mathbf{x}, \alpha, \beta)}$$

$$\text{where } s(\mathbf{x}, \alpha, \beta) = \alpha + \sum_{l=1}^L x_l \beta_l.$$

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

$$c'' = c' - 2 \log(\exp(\alpha) + \exp(-\alpha)) -$$

$$-4 \sum_{l=1}^L \log(\exp(\beta_l) + \exp(-\beta_l)).$$

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

$$\begin{aligned} & \log p_{\text{LogR}}(y_k | \mathbf{x}, \alpha, \beta) + c(\alpha, \beta) = \\ & = \sum_{n=1}^{N_{\mathcal{D}}} \log p(y_k^n | \mathbf{x}^n, \alpha, \beta) - 2|\alpha| - 4 \sum_{l=1}^L |\beta_l|. \end{aligned}$$

²Possible other choices include tree-augmented Naive Bayes (TAN) [Friedman *et al.*, 1997; Roos *et al.*, 2005; Greiner and Zhou, 2002].

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

We do this by simple line search, one parameter at a time³.

Finally, we apply 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

$$\sum_{n=1}^{N_D} \log w_k p(y_k^n | \mathbf{x}^n, \alpha, \beta) + c(\alpha, \beta) \quad (14)$$

where the weight w_k is the prior $p(y_k)$ divided by the observed relative frequency $\#\{n : y_k^n = y_k\} / N_D$.

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 2⁴.

5.1 Experimental Setup

For the gas flow of the diesel engine there is physical model from which a set of 29 residuals are automatically generated using structural analysis [Einarsson and Arrhenius, 2004; 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

³For larger problems faster methods, as for example discussed in [Minka, 2003] could be more suitable.

⁴The probabilities do not sum to one, since the probabilities for multiple faults are not included.

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.

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 contain 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 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.

Table 4 compares the logistic scores of the predictions given for the single faults by DI and LogR+weights. Note that because of inequality (9) 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

Table 4: Comparison 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

training data. This gives evidence for explaining away being important in this problem. Figure 3, in which the probabilities for each fault using LogR + weights are plotted, shows this in more detail. In the Figure 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.

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 what we expect to see in the real-world setting. In particular, evaluation data consists partly of previously unseen fault patterns. In addition there is prior knowledge available about which faults 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 to perform 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. Furthermore, it is crucial to include background information in the learning phase whenever it is available.

In our work to come we will apply the methods do several different applications in diagnosis to study if the results presented here are general. We will investigate models capable of both explaining away and taking prior knowledge into account, while providing an efficient inference procedure, as on-board computers offer very limited resources. We expect further improvement of performance is possible.

7 Acknowledgments

This project is partly financed by Scania CV AB and the Intelligent Vehicle Safety System (IVSS) program in Sweden.

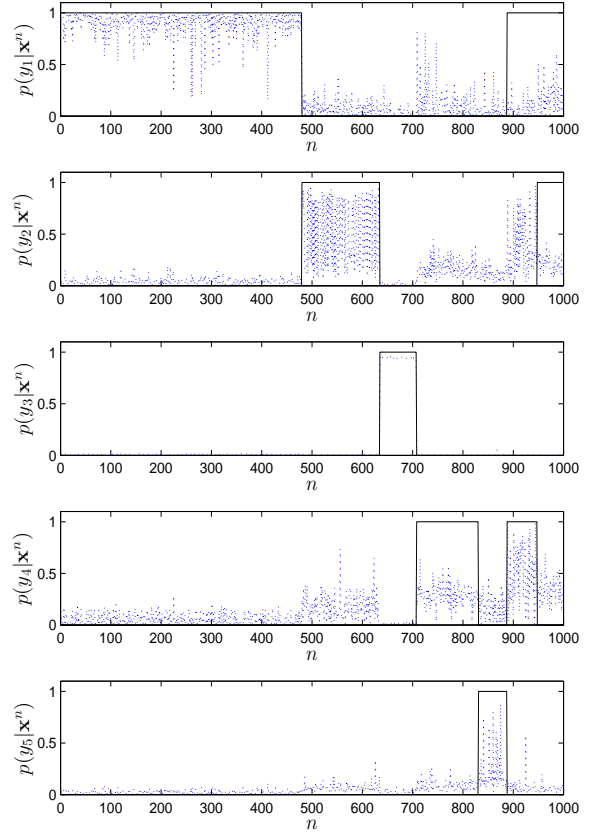


Figure 3: The predicted probability for the different faults given by LogR+w. Evaluation data is ordered after their fault patterns. The true fault is marked with a solid line.

References

- [Bishop, 1995] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [de Kleer and Williams, 1992] Johan de Kleer and Brian C. Williams. Diagnosis with Behavioral Modes. In *Readings in Model-based Diagnosis*, pages 124–130, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.
- [Devroye *et al.*, 1996] Luc Devroye, Laszlo Györfi, and Gabor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, New York, 1996.
- [Einarsson and Arrhenius, 2004] Henrik Einarsson and Gustav Arrhenius. Automatic design of diagnosis systems using consistency based residuals. Master’s thesis, Uppsala University, 2004.
- [Friedman *et al.*, 1997] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian Network Classifiers. *Machine Learning*, pages 131–163, 1997.
- [Greiner and Zhou, 2002] Russel Greiner and Wei Zhou. Structural Extension to Logistic Regression: Discrimina-

- tive Parameter Learning of Belief Net Classifiers. In *13th international conference on uncertainty in artificial intelligence*, 2002.
- [Hamscher *et al.*, 1992] Walter Hamscher, Luca Console, and Johan deKleer. *Readings in Model-based Diagnosis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.
- [Hartigan, 1975] John A. Hartigan. *Clustering Algorithms*. Wiley, 1975.
- [Heckerman *et al.*, 1995a] David Heckerman, John S. Breese, and Koos Rommelse. Decision-theoretic troubleshooting. *Communications of the ACM*, 38(3):49–57, 1995.
- [Heckerman *et al.*, 1995b] David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20(3):197–243, 1995.
- [Jensen, 2001] Finn V. Jensen. *Bayesian Networks*. Springer-Verlag, New York, 2001.
- [Kontkanen *et al.*, 2001] Petri. Kontkanen, Petri. Myllymäki, and Henry. Tirri. Classifier learning with supervised marginal likelihood. In J. Breese and D. Koller, editors, *Proceedings of the 17th International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 277–284, 2001.
- [Korbicz *et al.*, 2004] Jozef Korbicz, Jan M. Koscielny, Zdzislaw Kowalczyk, and Wojciech Cholewa. *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Springer, Berlin, Germany, 2004.
- [Krysander *et al.*, 2008] Mattias Krysander, Jan Åslund, and Mattias Nyberg. 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, 2008.
- [Langley and Sage, 1994] Pat Langley and Stephanie Sage. Induction of Selective Bayesian Classifiers. In *Proceedings of the 10th Conference on Uncertainty Artificial Intelligence*, 1994.
- [Lee *et al.*, 2007] Gareth Lee, Parisa Bahri, Srinivas Shastri, and Anthony Zaknich. A multi-category decision support framework for the tennessee eastman problem. In *Proceedings of the European Control Conference 2007*, Greece, 2007.
- [Lerner *et al.*, 2000] Uri Lerner, Ronald Parr, Daphne Koller, and Gautam Biswas. Bayesian Fault Detection and Diagnosis in Dynamic Systems. In *AAAI/IAAI*, pages 531–537, 2000.
- [Minka, 2003] Thomas P. Minka. A comparison of numerical optimizers for logistic regression. Technical report, Microsoft Research, 2003.
- [Mitchell, 1997] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
- [Narasimhan and Biswas, 2007] Sriram Narasimhan and Gautam Biswas. Model-based Diagnosis of Hybrid Systems. *IEEE Trans. on Systems, Man, and Cybernetics, Part A*, 37(3):348–361, 2007.
- [Ng and Jordan, 2002] Andrew Y. Ng and Michael I. Jordan. On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems 14*, 2002.
- [Nyberg, 2005] Mattias Nyberg. Model-Based Diagnosis of an Automotive Engine Using Several Types of Fault Models. *IEEE Transactions on Control Systems Technology*, 10(5):679–689, 2005.
- [Pernestål and Nyberg, 2007] Anna Pernestål and Mattias Nyberg. Diagnosing Known and Unknown Faults from Incomplete Data. In *Proceedings of European Control Conference*, 2007.
- [Pernestål *et al.*, 2006] Anna Pernestål, Mattias Nyberg, and Bo Wahlberg. 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, 2006.
- [Pulido *et al.*, 2005] Belarmino Pulido, Vicenc Puig, Theresa Escobet, and Joseba Quevedo. 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)*, 2005.
- [Reiter, 1992] Raymond Reiter. A Theory of Diagnosis From First Principles. In *Readings in Model-based Diagnosis*, pages 29–48, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.
- [Rish, 2001] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, 2001.
- [Roos *et al.*, 2005] Teemu Roos, Hannes Wettig, Peter Grünwald, Petri Myllymäki, and Henry Tirri. On Discriminative Bayesian Network Classifiers and Logistic Regression. *Machine Learning*, pages 267–296, 2005.
- [Roychoudhury *et al.*, 2006] Indranil Roychoudhury, Gautam Biswas, and Xenofon Koutsoukos. A Bayesian Approach to Efficient Diagnosis of Incipient Faults. In *Proceedings of 17th International Workshop on Principles of Diagnosis (DX 06)*, pages 243–250, 2006.
- [Russell and Norvig, 1995] S. Russell and Peter Norvig. *Artificial Intelligence*. Xxxx, 1995.
- [Schwall and Gerdes, 2002] Matthew Schwall and Christian Gerdes. A probabilistic Approach to Residual Processing for Vehicle Fault Detection. In *Proceedings of the 2002 ACC*, pages 2552–2557, 2002.
- [Sheppard and Kaufman, 2005] John W. Sheppard and Mark A. Kaufman. A Bayesian Approach to Diagnosis and Prognosis Using Built-In Test. *IEEE Transactions on Instrumentation and Measurement*, 54:1003–1018, 2005.
- [Silander and Myllymäki, 2006] Tomi Silander and Petri Myllymäki. A Simple Approach for Finding the Globally Optimal Bayesian Network Structure. In *Proceedings of the 22nd Conference on Uncertainty in AI (UAI)*, 2006.