A Bayesian Approach to Learning in Fault Isolation

Hannes Wettig
Helsinki Institute for
Information Technology
Finland
wettig@hiit.fi

Anna Pernestål
Dept. Electrical Engineering
Linköping University
Sweden
annap@isy.liu.se

Tomi Silander
Helsinki Institute for
Information Technology
Finland
tsilande@hiit.fi

Mattias Nyberg
Scania CV AB
Södertälje
Sweden
mattias.nyberg@scania.com

Abstract

Fault isolation is the art 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 fault isolators. The number of data, as well as the available computing resources, are often limited and there may be previously unobserved fault patterns. To meet these challenges we take on a Bayesian approach. We compare five different methods for learning in fault isolation, and evaluate their performance on a real fault isolation problem; the diagnosis of an automotive engine.

1 INTRODUCTION

We consider the problem of fault isolation, i.e. the problem 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 the current work we investigate and compare different methods for learning from training data and prior knowledge.

We are motivated by the problem of fault isolation in an automotive engine, and the learning methods are evaluated using experimental training data and evaluation data from real driving situations. In engine fault isolation there may be several hundreds of faults and observations. There will be fault patterns, i.e. cooccuring faults, from which there are no training data. Furthermore, training data is typically experimental and obtained by implementing faults, running the process, and collecting observations. On the other hand, there is often engineering knowledge available about the process. The engineering knowledge can for example be used to determine the structure of dependencies between faults and observations. This kind of knowledge is often the only basis in previous algorithms for fault isolation [6, 12, 19].

Due to the fact that there are previously unobserved fault patterns in training data, frequentist and purely data-based methods are bound to fail. To meet these challenges we use a Bayesian approach to learning in fault isolation. We consider five different methods of learning a model from training data, which are all previously present in the literature in different forms. We taylor 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 contributions of the current work are the investigation of Bayesian learning methods and regression models for fault isolation by comparing the five methods mentioned above, the application and evaluation of the methods on real-world data, and the combination of data-driven learning and prior knowledge within these methods. In order to do this investigation, we first discuss the characteristics of the fault isolation problem in terms of probability theory, and 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. Finally, we compare the five methods, and discuss their advantages and drawbacks.

Bayesian methods for fault isolation are previously studied in literature. In these previous works it is generally assumed that the model is given [26, 15], or can be derived from a physical model without using training data [17, 25]. In the current work on the other hand, we focus on *learning* the models. Previous works on Learning models for fault isolation typically rely on pattern recognition methods described e.g. in

[1, 3]. Examples of such methods are presented for example in [14]. Pattern recognition methods are applicable if there is sufficient training data available. Unfortunately, this is rarely the case in fault isolation. In [20] the problem of learning with missing fault patterns is discussed. In [20] training data is combined with fundamental methods for fault isolation described in [2, 22]. This approach is referred to as Direct Inference in the current work, and compared to the other four methods for learning.

The paper is structured as follows. We introduce notation, and formulate the diagnosis problem in Section 2. Therein we also define relevant performance measures. In Section 3 we briefly describe the five methods used, and in particular how they are applied to the diagnosis problem, before we perform the evaluating experiments and compare the results obtained in Section 4. Finally, in Section 5 we conclude the paper by summarizing our results and discussing future work directions.

2 PROBLEM FORMULATION

Before going into the details of each of the learning methods we introduce some notation, and discuss the characteristics of the fault isolation problem. Then we carefully state the problem at hand and define performance measures.

2.1 BAYESIAN FAULT ISOLATION

The fault isolation problem can be formulated as a prediction problem, where the task is to determine the fault(s) present in a system, given a set of observations from the system. Let the faults be represented by the binary variables $\mathbf{Y} = (Y_1, \ldots, Y_K)$, and let the observations from the system 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 $p(\mathbf{X} = \mathbf{x})$ (or simply $p(\mathbf{x})$) to denote either probabilities or probability distributions both in the continuous and in the discrete case. The meaning will be clear from the context.

We are given a set of training data $\mathcal{D}$, consisting of samples $(\mathbf{y}^n, \mathbf{x}^n)$, $n = 1, \ldots, N_\mathcal{D}$, 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 system we use a set $\mathcal{E}$ consisting of $N_\mathcal{E}$ samples. The evaluation data is collected by running the system, meaning that it is *observational.* Furthermore, we assume that the fault isolation algorithm is triggered by a fault detector telling us there must be *at least one fault present* in the process.

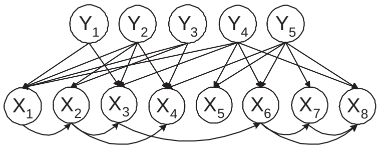

Figure 1: A Bayesian network describing a typical fault isolation problem.

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 assumed to be a priori independent, i.e. that

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

meaning that faults cannot 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 [6], 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 may 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 simply modeled with dependencies between observation variables. This is more carefully discussed in [21].

We take a Bayesian view point on fault isolation. The objective is to find the probability for each fault to be present given the current observation, the training data, and the prior knowledge I, i.e. to compute the probabilities $p(y_k|\mathbf{x}, \mathcal{D}, I)$, $k = 1, \ldots, K$. The probability for each fault can be found by marginalizing over $\mathbf{y}_{-k} = (y_1, \ldots, y_{k-1}, y_{k+1}, \ldots, y_K)$,

$$p(y_k|\mathbf{x}, \mathcal{D}, I) = \sum_{\mathbf{y}_{-k}} p(\mathbf{y}_{-k}, y_k|\mathbf{x}, \mathcal{D}, I). \qquad (2)$$

Note that $(\mathbf{y}_{-k}, y_k) = \mathbf{y}$, and (2) means that we seek the conditional distribution $p(\mathbf{y}|\mathbf{x}, \mathcal{D}, I)$. To simplify the notation we will omit the background information I in the equations.

Computing the conditional distribution $p(\mathbf{y}|\mathbf{x}, \mathcal{D})$ is generally difficult. To approximate it we need a model $\mathcal{M}$ and a method for determining the parameters of the model.

2.2 PERFORMANCE MEASURES

To evaluate the different models to be used in Bayesian fault isolation, we use two performance measures: log-score and percentage of correct classification.

The log-loss is a commonly used measure [1], and given by

$$\mu(\mathcal{E}, \mathcal{M}) = \frac{1}{N_{\mathcal{E}}} \sum_{j=1}^{N_{\mathcal{E}}} \log p(\mathbf{y}^j | \mathbf{x}^j, \mathcal{M}), \tag{3}$$

The scoring function μ measures two important properties of the fault isolation system; both the ability to assign large probability mass to faults that are present, and also 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 corresponds to the empirically observed probabilities. In the fault isolation problem the conditional probabilities for faults is often combined with decision theoretic methods for troubleshooting [8], where optimal decision making requires conditional probabilities close to the generating distribution.

The second measure we use is not proper. It is closely related to the 0/1-loss used e.g. in pattern classification [1]. However, in case of multiple faults present it suffices to assign highest probability to any of them. We define

$$\nu(\mathcal{E}, \mathcal{M}) = \#\{j : y^j_{max}(\mathbf{x}^j, \mathcal{M}) = 1\}/N_{\mathcal{E}}, \tag{4}$$

where $y^j_{max}(\mathbf{x}^j, \mathcal{M})$ is the fault assigned highest probability by $\mathcal{M}$ given $\mathbf{x}^j$. The ν-score reflects the performance of the fault isolation system combined with the simple troubleshooting strategy "check the most probable fault first".

3 MODELLING APPROACHES

In this section we briefly present the inference methods used to tackle the fault isolation problem. We carefully state all assumptions made, and describe the adjustments of each method to apply it to the diagnosis problem. However, we begin by describing two assumptions that need to be made for all methods except DI.

3.1 MODELLING 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. A priori this corresponds to approximation (1). However, when we build separate models for each fault, we also make a stronger assumption, namely that the faults *remain* independent given the observations,

$$p(\mathbf{y}|\mathbf{x}) = \prod_{k=1}^{K} p(y_k|\mathbf{x}, y_1, \ldots, y_{k-1}) \approx \prod_{k=1}^{K} p(y_k|\mathbf{x}) \tag{5}$$

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

$$\prod_{k=1}^{K} p(\mathbf{x}|y_k) \approx \prod_{k=1}^{K} p(\mathbf{x}|y_1, \ldots, y_k), \tag{6}$$

meaning 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"* [10]. Looking at Figure 1 we observe, that this indeed is a strong assumption, since there are unshielded colliders (V-structures, bastards, common children of non-connected nodes) of the faults present.

Assumption (5) is primarily made for technical reasons, in order to be able to build separate models for each fault. But often it is also the case (as in the application of Section 4) that there is training data only from single faults. This means we do not have any training data telling us about the joint effect of multiple faults.

Remember that it is known that there is at least one fault present when the fault isolator is employed, see Section 2.1. Therefore, instead of computing $p(\mathbf{y}|\mathbf{x})$, we search

$$p(\mathbf{y}|\mathbf{x}, \sum_k y_k > 0) = p(\mathbf{y}|\mathbf{x})(1 - p(\mathbf{y} \equiv \mathbf{0}|\mathbf{x})). \tag{7}$$

Unfortunately

$$p(\mathbf{y}|\mathbf{x}, \sum_k y_k > 0) \neq \prod_k p(y_k|\mathbf{x}, \sum_k y_k > 0), \tag{8}$$

a fact which recouples the single-fault models introduced in (5). This fact is ignored during the learning phase and the single-fault models are trained individually. We then apply (7) in the evaluation phase.

3.2 DIRECT INFERENCE

Several previous fault isolation algorithms rely on prior knowledge about which observations may be affected

Table 1: An example of an FSM

	Y_1	Y_2	Y_3
X_1	1	1	0
X_2	1	0	1

by each fault [2, 22, 12]. 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. The direct inference method aims at combining the information given by the FSM with the training data available. Assume that observations are binary and that the background information I containing the FSM is given. Then, under certain assumptions it can be shown [20] that

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}) = \begin{cases} 0 & \mathbf{x} \in \gamma \\ \frac{n_{\mathbf{xy}}+\alpha_{\mathbf{xy}}}{N_{\mathbf{y}}+A_{\mathbf{y}}} \frac{p(\mathbf{y}|I)}{\pi_0} & \text{otherwise,} \end{cases} \tag{9}$$

where π_0 is a normalization constant, $n_{\mathbf{xy}}$ is the count of training data with fault $\mathbf{y}$ and observations $\mathbf{x}$, $\alpha_{\mathbf{xy}}$ is a parameter describing the prior belief in the observation $\mathbf{x}$ when the fault is $\mathbf{y}$ (a *Dirichlet* prior), $N_y = \sum_{\mathbf{x}'} n_{\mathbf{x}'\mathbf{y}}$, and $A_y = \sum_{\mathbf{x}'} \alpha_{\mathbf{x}'\mathbf{y}}$. The sets γ are determined by the background information as described in [20].

The direct inference 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.

3.3 BAYESIAN NETWORKS

When using Bayesian networks for prediction, we search 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 $\mathbf{y}$ can be computed. We consider two types of Bayesian networks: Naive Bayes and general Bayesian Networks.

3.3.1 Naive Bayes

The Naive Bayes classifier assumes that the observations are independent given the fault. Naive Bayes is is one of the standard methods for Bayesian prediction and often performs surprisingly well [3, 23]. However, due to the erroneous independence assumptions it is poorly calibrated when there are strong dependencies between the observations. To alleviate this problem, we apply variable selection according to an internal leave-one-out scoring function:

$$S(V) = \frac{1}{N_{\mathcal{D}}} \sum_{n=1}^{N_{\mathcal{D}}} \log P(y_k^n|\mathbf{x}^n, V, \mathcal{D} \setminus \{(\mathbf{y}^n, \mathbf{x}^n)\}, \alpha), \tag{10}$$

where $V \subset \mathbf{X}$ is the variable set under consideration and α is the Dirichlet hyper-parameter for the NB-model.

3.3.2 General Bayesian Network

Since it is known that the faults causally precede the observations, and since the observations are known to be dependent given the faults, a natural step forward from the Naive Bayes structure is a Bayesian network. In the network we constrain the fault to be a root node, but otherwise leave the structure unconstrained. One such network was learned for each fault using a BDe score (with an equivalent sample size parameter of 1.0). For small systems (< 30 variables) learning can be performed using the exact algorithm in [27], while for larger systems approximate methods, e.g. [9], can be used.

3.4 REGRESSION

Fault isolation is a discriminative task, where we are to predict the fault vector $\mathbf{y}$ given the observations $\mathbf{x}$, i.e. estimate the conditional likelihood

$$p(\mathbf{y}|\mathbf{x}, \theta) = \frac{p(\mathbf{y}, \mathbf{x}|\theta)}{\sum_{\mathbf{y}} p(\mathbf{y}, \mathbf{x}|\theta)}. \tag{11}$$

It is well known [18, 11] 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 3.3. Regression models form a family of such methods.

3.4.1 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

$$p(y_k|\mathbf{x}) = \frac{1}{Z} \exp(-\frac{(\mathbf{w}_k^T \mathbf{x} + w_{k0} - y_k)^2}{2\sigma^2}), \tag{12}$$

where Z is a normalization constant. To determine the parameters we use the standard methods described for example in [1].

$$\begin{aligned}\mathbf{w}^* &= \arg\min_{\mathbf{w}} -\sum_{n=1}^{N_{\mathcal{D}}} \log p(y_k^n|\mathbf{x}^n, \mathbf{w}) \\ &= \arg\min_{\mathbf{w}} -\sum_{n=1}^{N_{\mathcal{D}}} (\mathbf{w}_k^T \mathbf{x}^n + w_{k0} - y_k^n)^2.\end{aligned}$$

When the parameters $\mathbf{w}^*$ are known, the parameter σ can also be computed. The normalization constant in (12) is given by $Z = \exp(-((\mathbf{w}^*)_k^T\mathbf{x}+w_{k0}^*-1)^2/2\sigma^2)+\exp(-((\mathbf{w}^*)_k^T\mathbf{x}+w_{k0}^*-0)^2/2\sigma^2)$.

3.4.2 Logistic Regression

Learning parameters to maximize (11) for a Bayes Net $\mathcal{B}$ is known to be equivalent to *logistic regression* under the condition that no child of the class can be a "bastard", a common child of two variables that are not interconnected directly. More formal definition and proofs can be found in [24]. In our case, this implies approximation (5).

To start with, for each fault we learn a logistic regression model corresponding to a discriminative Naive Bayes classifier [1].

We name the parameters of the logistic regression model α and β such that the conditional likelihood is defined as

$$p(y_k = 1|\mathbf{x}, \alpha, \beta) := \frac{\exp s(\mathbf{x}, \alpha, \beta)}{\exp s(\mathbf{x}, \alpha, \beta) + \exp -s(\mathbf{x}, \alpha, \beta)} \tag{13}$$

where

$$s(\mathbf{x}, \alpha, \beta) := \alpha + \sum_{l=1}^{L} x_l \beta_l. \tag{14}$$

We also include a smoothing term $c(\alpha, \beta)$ in our objective function which takes the place of a prior in the corresponding NB classifier. To unify its role for different observations, we first normalize our data by shifting and scaling such that for $l = 1, \ldots, L$

$$\sum_n x_l^n = 0 \quad \text{and} \quad \max_n |x_l^n| = 1 \tag{15}$$

Starting out from the uniform prior, we pretend to have seen one vector of each class at node Y_k and two vectors of each class with extreme values ±1 at each node X_l, with all other values zero (∼unobserved). This amounts to a smoothing term

$$\begin{aligned}c'(\alpha,\beta) &- 2\log(\exp(\alpha)+\exp(-\alpha)) \\ &- 4\sum_{l=1}^{L}\log(\exp(\beta_l)+\exp(-\beta_l)).\end{aligned} \tag{16}$$

However, we found this smoothing term problematic, since it is flat near zero. Therefore, we never get any parameters exactly zero. But in logistic regression many small parameters can make a difference, while they may be weakly supported. We choose to replace $\log(\exp(x) + \exp(-x))$ by $|x|$. 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 as to maximize

$$\begin{aligned}&\log p(\mathbf{y}_k|\mathbf{x}, \alpha, \beta) + c(\alpha, \beta) \\ &= \sum_{n=1}^{N_{\mathcal{D}}} \log p(y_k^n|\mathbf{x}^n, \alpha, \beta) - 2|\alpha| - 4\sum_{l=1}^{L}|\beta_l|.\end{aligned} \tag{17}$$

We do this by simple line search, one parameter at a time[2].

Finally, we try a variant of this algorithm which weights the training vectors. We have prior knowledge about the probabilities $p(y_k)$ with which to expect some fault y_k in the real-world setting or, in this case, the evaluation set. These probabilities differ from the relative frequencies observed in the training set. The idea is to weight the training vectors in the objective as to focus the optimization on areas of the data space more likely to be seen later on. The corresponding objective for fault Y_k becomes

$$\sum_{n=1}^{N_{\mathcal{D}}} \log w_k p(y_i^n|\mathbf{x}^n, \alpha, \beta) + c(\alpha, \beta) \tag{18}$$

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

4 EXPERIMENTS

To evaluate the different methods learning fault isolation models, 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 in Table 2.

[1] possible other choices include tree-augmented Naive Bayes (TAN) [24, 5]

[2] There are much faster optimization techniques, some of which are compared in [16], but for our purposes this did nicely

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

4.1 EXPERIMENTAL SETUP

For the gas flow of the diesel engine there is physical model from which a set of 29 diagnostic tests are automatically generated using structural analysis [4, 13]. Each of the observations is constructed to be sensitive to a subset of the faults.

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 all except the regression algorithms take in discrete data. The data is discretized in two different ways: binary, with thresholds set such that all fault free data is known to be contained in the same bin; and discretized using k-means clustering [7] 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 3 the NB and DI algorithms perform best if not all observations are used. For both DI and NB we perform variable selection such that an internal log-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.

4.2 RESULTS

In Table 3 the log-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	log-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

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

We observe, 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 log-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 Bayes Nets the conditional probability tables easily grow too large. 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 become smaller. We observe that this score favors the regression models and the Bayesian methods using binary data. The reason for the good performance of the methods using binary data is the particular way of thresholding the data such that all fault free samples are contained in the same bin.

Table 4 compares the log-scores of the predictions given for the single faults by DI and LogR+weights. Note that because of inequality (8) the columns do not sum to the corresponding entries in Table 3. Not surprisingly, both methods (as all others) have most trouble with 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 2, 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 rightmost samples have multiple faults, visualizing that the double faults are most difficult to predict.

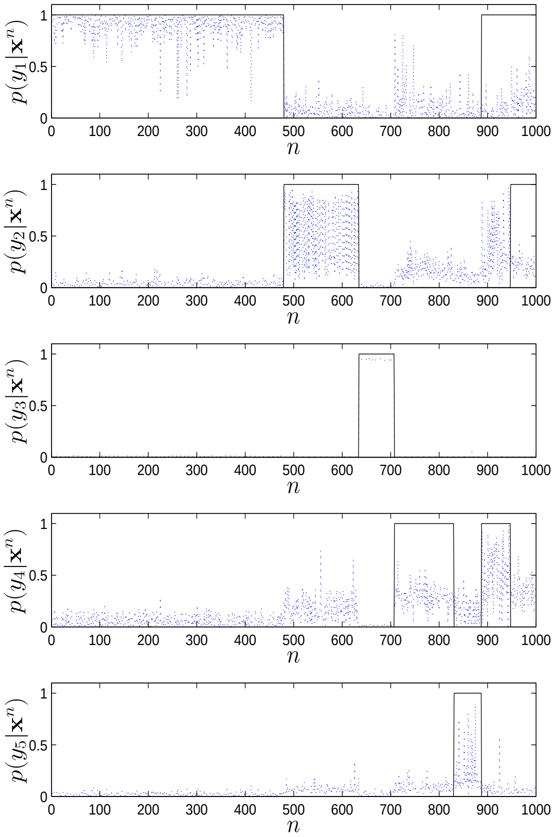

Figure 2: The predicted probability for the different faults given by LogR+w. Evaluation data is ordered after their fault patterns. The true fault is marked with a solid line.

5 CONCLUSIONS

We have considered the problem of fault isolation in an automotive diesel engine. We have 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 the discriminative logistic regression method to perform best. Among the best methods 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. All the methods used, except direct inference, ignore explaining away. 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 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.

References

[1] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[2] 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.

[3] Luc Devroye, Laszlo Györfi, and Gabor Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer, New York, 1996.

[4] Henrik Einarsson and Gustav Arrhenius. Automatic design of diagnosis systems using consistency based residuals. Master's thesis, Uppsala University, 2004.

[5] Russel Greiner and Wei Zhou. Structural Extension to Logistic Regression: Discriminative Parameter Learning of Belief Net Classifiers. In *13th international conference on uncertainty in artificial intelligence*, 2002.

[6] Walter Hamscher, Luca Console, and Johan deKleer. *Readings in Model-based Diagnosis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

[7] John A. Hartigan. *Clustering Algorithms*. Wiley, 1975.

[8] David Heckerman, John S. Breese, and Koos Rommelse. Decision-theoretic troubleshooting. *Communications of the ACM*, 38(3):49–57, 1995.

[9] 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.

[10] Finn V. Jensen. *Bayesian Networks*. Springer-Verlag, New York, 2001.

[11] 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.

[12] Jozef Korbicz, Jan M. Koscielny, Zdzislaw Kowalczuk, and Wojciech Cholewa. *Fault Diagnosis. Models, Artificial Intelligence , Applications.* Springer, Berlin, Germany, 2004.

[13] 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.

[14] 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.

[15] Uri Lerner, Ronald Parr, Daphne Koller, and Gautam Biswas. Bayesian Fault Detection and Diagnosis in Dynamic Systems. In *AAAI/IAAI*, pages 531–537, 2000.

[16] Thomas P. Minka. A comparison of numerical optimizers for logistic regression. Technical report, Micrsoft Research, 2003.

[17] 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.

[18] 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.

[19] 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.

[20] Anna Pernestål and Mattias Nyberg. Diagnosing Known and Unknown Faults from Incomplete Data. In *Proceedings of European Control Conference*, 2007.

[21] 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.

[22] 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.

[23] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, 2001.

[24] 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.

[25] 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.

[26] 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.

[27] 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.