Bayesian Solutions to Control Loop Diagnosis

by

Ruben Gonzalez

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Process Control

Department of Chemical and Materials Engineering
University of Alberta

Abstract

While there has been much literature in the area of system monitoring and diagnosis, most of these techniques have a relatively small scope in terms of the faults and performance issues that they are built to detect. When implementing several monitors simultaneously on a single process, a single problem can result in multiple alarms, making it difficult to single out the underlying cause. Recent work has been done on incorporating information from multiple monitoring systems by means of Bayesian diagnosis; however, work so far is still in its infancy. This thesis focuses on a number of techniques that can be used to improve performance of previously proposed Bayesian diagnosis techniques.

Previous work [1] improved Bayesian diagnosis by accounting for incomplete evidence (monitor readings). Evidence is often presented in a multivariate vector, thus evidence with missing elements is incomplete. Missing elements can also appear in the mode (or set of problem sources). Many times, the mode information can also be incomplete within the historical data, such modes are *ambiguous*. This thesis develops two approaches for handling ambiguous modes. One technique is derived using Bayesian methods, while another technique is a modification on Dempster-Shafer Theory.

Evidence in previous work [2] [3] was considered to be a vector of discrete variables, and the resulting probability estimates consisted of discrete categorical distributions. However, most monitors have continuous outputs that are only discretized for the sake of alarms. Discretization results in information loss, so it is desirable to use a technique that can easily estimate likelihoods for continuous evidence. Kernel density estimation is a popular technique for the non-parametric estimation of probability densities. Non-paramteric methods enjoy the advantage of not requiring assumptions on the nature of the distribution, so that they naturally fit the shape of the data's distribution (which is the main motivation for discretization). Kernel density estimation enables the construction of non-parametric estimates for continuous densities, allowing us to circumvent discretization procedures.

Bootstrapping was a topic of interest for generating additional data if the data was sparse; however, it is also likely that modes will be sparse, that is, the history will often not

contain all modes of interest. This thesis presents a two-pronged approach: first, to break down the problem into analysing components and properly selecting monitors; second, to generate additional modes by incorporating gray-box models and bootstrapping.

Finally, incorporating ambiguous modes will affect the autocorrelated mode solution [4], while incorporating continuous evidence through kernel density estimation will affect the autocorrelated evidence solution [5]. This thesis lays down a framework for dynamic implementation of the newly proposed ambiguous mode and continuous evidence techniques.

To my beloved wife Ellanor, and to God who have both been the greatest sources of comfort and encouragement while I was completing this work

Acknowledgements

First and foremost, I would like to thank my supervisor Dr. Huang for his continuing advice support and guidance, in particular, his guidance in how to present some of the more technically difficult components of this thesis. I would also like to thank him for his advice toward many of the areas that became major contributions in this thesis, as well as his patience during the times when I had difficulty in understanding and communicating some of these concepts.

I would also like to acknowledge Fei Qi, a previous doctoral student whose work I have been following. He has been of great help in terms of understanding his work, and the suggestions for future research have been particularly helpful (namely the use of continuous evidence and dealing with modes not represented in the data).

A significant amount of this work was developed as a direct result from industrial collaboration. I would like to thank Aris Espejo and Joseph Amalraj from Syncrude for their help in acquiring the industrial data and process knowledge which was used prior to this thesis. Additionally, I would like to thank Elom Domlan from Suncor for his help in acquiring data that was directly used in this thesis, and for his guidance in assessing underlying process modes. Finally, I would like to thank Eric Lau and Ramesh Kadali from Suncor for their encouragement and guidance in applying some of the material in this thesis. Through working with them, I have gained a stronger appreciation of the practical benefits of the methodologies I have been developing during my time as a graduate student.

I would also like to thank the members of my research group, particularly Kangkang Zhang, Ming Ma, Tianbo Lu, Shima Khatibisepehr, Yaojie Lu, Yuri Shardt, Xin Jing, Yijia Zhu who have all made specific contributions to my work, but also to the other group members who have also contributed in many indirect ways. I would also like to aknowledge the department of Chemical and Materials Engineering for giving me the opportunity to pursue my doctoral degree, and I would like to thank NSERC, Alberta Scholarship Programs, Syncrude Ltd, and Suncor for their financial support.

I would like to thank my parents for their continuous encouragement and for always being so enthusiastic and supportive of the decisions I've made. They may have been at the other end of the country, but often times it did not feel that way. Finally, I would like to thank Ellanor, now my wife, who has been such a great friend to me while we were living in opposite ends of the country. Her heartfelt support was crucial for me during the difficult

beginnings of this work, and I very much doubt that this thesis would have been completed without it. I would also like to thank Ellanor for her continuing patience as I worked to complete this thesis while we were together. She has honestly made the final years of my graduate studies the happiest years of my life.

Contents

List of Figures

List of Tables

List of Symbols

Symbol	Description
α	Frequency parameter for the Dirichlet Distribution
$\boldsymbol{\alpha}\{\frac{\bullet}{\boldsymbol{m}_k}\}$	Frequency parameters pertaining to the ambiguous mode $\boldsymbol{m}_k$
μ	Population mean
Σ	Population covariance
σ	Population standard deviation
Θ	Complete set of probability/proportion parameters
$\Theta\{\frac{\bullet}{\boldsymbol{m}_k}\}$	The set of elements in Θ pertaining to the ambiguous mode $\boldsymbol{m}_k$
$\hat{\Theta}$	Informed estimate of Θ
$\boldsymbol{\Theta}$	Complete set of probability/proportion parameters (matrix form)
$\boldsymbol{\Theta}^*$	Inclusive estimate of Θ (matrix form)
$\boldsymbol{\Theta}_*$	Exclusive estimate of Θ (matrix form)
θ	A probability/proportion parameter
$\theta\{\frac{m}{\boldsymbol{m}}\}$	Proportion of data in ambiguous mode $\boldsymbol{m}$ belonging to mode m
$Bel(M)$	Lower bound probability of mode M
C	State of the component of interest (random variable)
c	State of the component of interest (observation)
$\mathcal{C}(M)$	The event where mode M was diagnosed
$\mathcal{C}(M)\|M$	The event where mode M was diagnosed and M was true
$\mathcal{C}(\bar{M})\|M$	The event where a mode other than M was diagnosed and M was true
$\mathcal{D}$	Historical record of evidence
D_i	i^{th} element of historical evidence data record $\mathcal{D}$
E	Evidence (random variable)
e	Evidence (observation)
F_N	False negative diagnosis rate
$\boldsymbol{G}$	Generalized BBA

$\boldsymbol{G}[\mathbf{:}, m]$	m^{th} column of $\boldsymbol{G}$ (MATLAB notation)
$\boldsymbol{G}[k, \mathbf{:}]$	k^{th} row of $\boldsymbol{G}$ (MATLAB notation)
H	Bandwidth matrix (Kernel density estimation)
$\boldsymbol{H}$	Hessian matrix
IID	Independent and identically distributed
$\boldsymbol{J}$	Jacobian matrix
K	Support for conflict (Dempster-Shafer Theory)
K	Kernel function (Kernel density estimation)
M	Operational mode (random variable)
$\boldsymbol{M}$	Potentially ambiguous operational mode (random varaible)
m	Operational mode (observation)
$\boldsymbol{m}$	Potentially ambiguous operational mode (observation)
MIC	Mutual information criterion
$CMIC$	Conditional mutual information criterion
$n(E)$	Number of times evidence E has been observed
$n(E, M)$	Number of times evidence E and mode M have been jointly observed
$n(M)$	Number of times mode M was observed
ODE$[f(x)]$	Ordinary differential equation solver applied to $f(x)$
$p(E)$	Normalization over Evidence (probability of evidence)
$p(E\|M)$	Likelihood (probability of evidence given the mode)
$p(M)$	Prior (prior probability of the mode)
$p(M\|E)$	Posterior (probability of mode given the evidence)
$Pl(M)$	Upper bound probability of mode M
P	Posterior state covariance (Kalman filter)
Q	Model error covariance (Kalman filter)
R	Observation error covariance (Kalman filter)
S	Sample covariance matrix
$S(E\|\boldsymbol{M})$	Support for evidence E given potentially ambiguous mode $\boldsymbol{M}$
$S(\boldsymbol{M})$	Support for potentially ambiguous mode $\boldsymbol{M}$
$S(E\|\boldsymbol{M})$	Support for potentially ambiguous mode $\boldsymbol{M}$ given evidence E
UKF$[f(x)]$	Unscented Kalman filter with a model $f(x)$

Chapter 1

Introduction

1.1 Motivational Illustrations

Consider the following scenarios:

Scenario A

You are a plant operator, and a gas analyser reading triggers an alarm for a low level of a vital reaction component; however, from experience, you know that this gas analyser is prone to error. The difficulty is however, if the the vital reaction component is truly scarce, its scarcity could cause plugging and corrosion downstream that could cost over $ 120 million in plant downtime and repairs, but if the reagent was not low, shutting down the plant would result in $ 30 million in downtime. Now, imagine that we have a diagnosis system that has recorded several events like this in the past, using information from both upstream and downstream, and is able to generate a list of possible causes of this alarm reading, and displays the probability of each scenario. The diagnosis system indicates that the most possible cause was a scenario that happened three years ago, when the vital reagent concentration truly dropped, and by quickly taking action to bypass the downstream section of the plant, a 120-million-dollar incident was successfully avoided. Finally, imagine that you are the manager of this plant, and discover that after implementing this diagnosis system, the incidents of unscheduled downtime have been reduced by 60 % and that incidents of false alarms have been reduced by 80 %.

Scenario B

You are the head of a maintenance team of another section of the plant with over 40 controllers and 30 actuators. Oscillation has been detected in this plant, where any of these controllers or actuators could be the cause. Because these oscillations can push the system into risky operating regions, caution must be exercised to keep the plant in a safer region, but at the cost of poorer product quality. Now, imagine you have a diagnosis tool that has data recorded from previous incidents and their troubleshooting solutions, and

the probabilities of each incident. With this tool, we see that the most probable cause (at 45 %) was fixed by replacing the stem packing on Valve 23, and that the second most probable cause (at 22 %) was a tank level controller that in the past, was sometimes over-tuned by a poor application of tuning software. By looking at records, you find out that a young engineer recently used tuning software to re-tune the level controller. Because of this information, and because changing the valve packing costs more, you re-tune the controller during scheduled maintenance, and at start-up find that the oscillations are gone, and you can now safely move the system to a point that produces better product quality. Now that the problem has been solved, you update the diagnosis tool with the historical data to improve the tool's future diagnostic performance. Now imagine, that as the head engineer of this plant, you find out that 30 % of the most experienced people on your maintenance team are retiring this year, but because the diagnostic system has documented a large amount of their experience, new operators are better equipped to figure out where the problems in the system truly are.

Overview

These stories paint a picture of why there has been so much research interest in fault and control loop diagnosis systems in the process control community. The strong demand for better safety practices, decreased downtime, and fewer costly incidents (coupled with the increasing availability of computational power) all fuel this active area of research. Traditionally, a major area of interest has been in detection algorithms (or *monitors* as they will be called in this work) that focus on the behaviour of the system component. The end goal of implementing a monitor is to create an alarm that would sound if the target behaviour is observed. As more and more alarms are developed, it becomes increasingly probable that a single problem source will set off a large number of alarms, resulting in an alarm flood. Such scenarios in industry have caused many managers to develop alarm management protocols within their organizations. Scenarios such as those presented in scenarios A and B can be realized and in some instances have already been realized by research emphasizing the best use of information obtained from monitors and historical troubleshooting results.

1.2 Previous Work

1.2.1 Diagnosis techniques

The principal objective in this thesis is to diagnose the operational mode of the process, where the mode comprises of the operational state of all components within the process. For example, if a system comprises of a controller, a sensor and a valve, the mode would contain information about the controller (e.g. well tuned or poorly tuned), the sensor (e.g. biased or unbiased), and the valve (e.g. normal or sticky). As such, the main problem

presented in this thesis falls within the scope of *fault detection and diagnosis*.

Fault detection and diagnosis has a vast (and often times overwhelming) amount of literature devoted to it for two important reasons:

1. The problem of fault detection and diagnosis is a legitimately difficult problem due to the sheer size and complexity of most practical systems.

2. There is great demand for fault detection and diagnosis as it is estimated that poor fault management has cost the United States alone more than $20 Billion annually as of 2003 [6].

In a three-part publication, Venkatasubramanian et al. [7] review the major contributions to this area and classifies them under the following broad families: quantitative model-driven approaches [7], qualitative model-driven approaches [8], and process data-driven approaches [9]. Each type of approach has been shown to have certain challenges. Quantitative model-driven approaches require very accurate models that cover a wide array of operating conditions; such models can be very difficult to obtain. Qualitative model-driven approaches require attention to detail when developing heuristics, or else one runs the risk of a spurious result. Process data driven approaches have been shown to be quite powerful in terms of detection, but most techniques tend to yield results that make fault isolation difficult to perform. In this thesis, particular interest is taken in the quantitative model-driven approach and the process data-driven approaches.

Quantitative model-driven approaches

Quantitative model-driven approaches focus on constructing models of a process and using these models to diagnose different problems within a process [10] [11]. These techniques bear some resemblance to some of the monitoring techniques described in Section 1.2.2 applied to specific elements in a control loop. Many different types of model-driven techniques exist, and were broken down according to Frank [12] as follows:

1. *The Parity Space Approach*, which looks at analytical redundancy in equations that govern the system [13].

2. *The Dedicated Observer and Innovations Approach*, which filters residual errors from the Parity Space Approach using an observer [14].

3. *The Fault Detection Filter Approach*, which augments the State Space models with fault-related variables [15] [16]

4. *Parameter identification approach* which is traditionally performed offline [12]. Here, modelling techniques are used to estimate the model parameters, and the parameters themselves are used to indicate faults.

A popular subclass of these techniques are deterministic fault diagnosis methods. One popular method in this subclass, is the parity space approach [13] which set up parity equations having analytical redundancy to look at error directions that could correspond to faults. Another popular method the observer-based approach [17] which uses an observer to compare differences in the predicted and observed states.

Stochastic techniques, in contrast to the deterministic techniques, use fault-related parameters as augmented states; these methods enjoy the advantage of being less sensitive to process noise [18], being able to determine size and precise cause of the fault, but are very difficult to implement on large-scale systems and often require some excitement [19]. Including physical fault parameters in the state often requires a nonlinear form of the Kalman filter (such as the EKF, UKF or the Particle Filter) as states often have nonlinear relationships with respect to parameters. Such techniques were pioneered by Isermann [20] [21] with other important contributions coming from Rault et al. [22]. The main reason behind including these parameters in the state is that the stochastic Kalman filter is primarily focused on estimating the state distribution; when fault parameters are included in the state, fault parameter distributions are automatically estimated in parallel with the state. Examples of this technique include Gonzalez et al. [23] which made use of continuous augmented bias states, while Lerner et al. [24] made use of discrete augmented fault states.

Process data-driven approaches

A popular class of techniques for process monitoring are data-driven modelling methods, where one of the more popular techniques is Principal Component Analysis (PCA) [25]. These techniques create black-box models assuming that the data can be explained using a linear combination of independent Gaussian latent variables [26]; a transformation method is used to calculate values of these independent Gaussian variables, and abnormal operation is detected by performing a significance test. The relationship between abnormal latent variables and the real system variables is then used to help the user determine what the possible causes of abnormality could be. There have also been modifications of the PCA model to include multiple Gaussian models [27] [25] where the best local model is used to calculate the underlying latent variables used for testing.

All PCA models assume that the underlying variables are Gaussian, but more recent methods [28] do away with this assumption by first using ICA to calculate values of independent latent variables (which are not assumed to be Gaussian under ICA) and then use Kernel density estimation to evaluate the probability density of that value. Low probability densities indicate that the process is behaving abnormally. Even more recent work [29] uses Bayesian networks instead of PCA/ICA to break down the system into manageable pieces; this allows the user to define variables of interest for monitoring and determine the causal structures used to help narrow down causes. Abnormality is detected if key process variables take on improbable values or if groups of key process variables take on improbable patterns.

Results from this type or approach is generally easier to interpret than PCA/ICA-based methods.

Bayesian data-driven approaches

This thesis focuses on using the Bayeisan data-driven approach, which is distinct from other fault detection and diagnosis methods, mainly for the reason that the Bayesian approach is a *higher-level diagnosis method* [3] [2]. This type of approach is not meant to compete with previously mentioned fault detection and diagnosis methods; instead, the Bayesian approach provides a unifying framework to simultaneously use many of these methods at once. As such, it can take input from many different fault detection and diagnosis techniques in order to make a final decision. In this thesis, other diagnosis methods and even instruments themselves are treated as input sources and are referred to as *monitors*; this term is chosen mainly because previous work [3] focused heavily on using input from control loop monitoring techniques (described later in Section 1.2.2).

For Bayesian diagnosis, data from monitors must be collected for every scenario that one would wish to diagnose. In this work, such scenarios are referred to as operational *modes*. When new monitor information arrives, the new information is compared to historical data in order to determine which historical mode best fits the new information. The Bayesian diagnosis technique ranks each of the modes based on posterior probability which is calculated using Bayes' Theorem [30]

$$p(M|E) = \frac{p(E|M)p(M)}{p(E)}$$
$$P(E) = \sum_i p(E|m_i)p(m_i)$$

where

- $p(M|E)$ is the posterior probability, (probability of the mode M given evidence E)
- $p(E|M)$ is the likelihood of the evidence E given the historical mode M
- $p(M)$ is the prior probability of the historical mode M
- $p(E)$ is the probability of the evidence E (which is a normalizing constant)

In the Bayesian diagnosis technique, the historical data and mode classifications are used to construct the likelihood $p(E|M)$, and prior probabilities of modes are assigned to $p(M)$ using expert knowledge. While collecting data for historical modes may be a challenge, the Bayesian method at least allows us to collect data in a way that is not necessarily representative of the true mode occurrence rate. For example, if mode 1 occurs 90 % of the time, then representative sampling would require that 90 % of the data come from mode 1. Bayesian methods (which use prior probabilities to cover mode representation) allow us to collect an arbitrary amount of data for each mode, giving us a lot more flexibility in data collection than other methods.

1.2.2 Monitoring techniques

Much of this work focuses on monitoring and diagnosing control-loops (a schematic for a typical control loop is given in Figure 1.1); for this area of research, there exists abundant literature on assessing the performance of the entire loop as well as diagnosing problems within the loop's core components. These methods can be directly used to create alarms or notification statuses which alert operators and engineers about risky or inefficient operation.

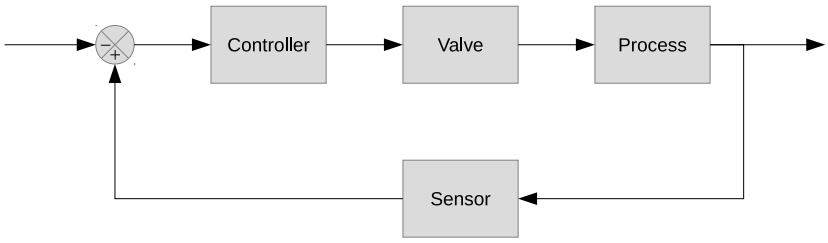


Figure 1.1: Typical control loop

Monitors used for control loops tend fall under the following categories:

- Control performance monitors
- Sensor bias monitors
- Valve stiction monitors
- Process model monitors
- Process operation monitors

Methods in this thesis are tested on three particular testbed systems: a simulated system, a bench scale system, and an industrial scale system. Each type of monitor has been used in at least one of the systems, a summary of monitors for each system is presented in Table 1.1. The simulated system makes use of three monitors (control performance monitors, valve stiction monitors, and process model monitors) while the bench scale system makes use of the two remaining monitor types (a process operation monitor, and two sensor bias monitors). The industrial-scale system uses no monitors, but instead, directly uses data from the various sensors within the facility.

Control performance benchmarks

Control performance monitors have the broadest scope of all the monitors that can be used in our applications, as they focus on the performance of the entire control loop. The output of a control performance monitor is a performance index based on some sort of ideal

Table 1.1: List of monitors for each system

Simulated	Bench Scale	Industrial Scale
Control Performance	Sensor Bias	Raw Sensor Readings
Valve Stiction	Process Operation	
Process Model		

benchmark. Ideally, a process monitor yields a value between 0 and 1 (for the worst and best performance respectively).

The first major control loop monitors were introduced by Harris [31] who proposed the *minimum variance control* (MVC) benchmark as a reference of evaluation; Huang et al. [32] later developed a filtering and correlation (FCOR) algorithm to effectively estimate this benchmark. The MVC benchmark was later extended to MIMO (multi-input multi-output) systems [33] along with the corresponding estimation algorithms [34]. In addition, other benchmarks were proposed, including the *linear quadratic regulator* benchmark [34] and the *model-predictive control* benchmark [35], [36], [37].

In this thesis, the FCOR algorithm [32] is the control performance monitor used for the simulated system.

Valve stiction monitors

As previously mentioned, fault detection and diagnosis using quantitative models is a broad area of research, and many techniques have been developed for specific applications based on knowledge of the physical system. Such techniques have been used for process control applications, where valve stiction monitoring is one of the most prominent modelling subjects, due to the fact that it is often the largest contributor to poor control loop performance. This subject has been reviewed in depth by Jelali and Huang [38]. Some of the earliest work in this area focused on non-linearity caused by non-ideal valves; such methods used a Hammerstein model to identify the non-linear valve and a linear process [39] [40]. Some qualitative methods were also introduced, for example, trying to fit the valve input-output relationship to an ellipse [41].

In this thesis, a simple stiction monitoring algorithm is used for the simulated system based on fitting the valve's input-output relationship to an ellipse. If stiction is absent, the data should be easily fit by an ellipse. However, if the fit is poor, it is likely that stiction is present.

Plant-model mismatch monitors

One of the main issues with model-predictive control is the gradual degradation of the process model. If the modelled behaviour deviates too drastically from the real process, the

efficacy of model-predictive control is fundamentally compromised. In order to safeguard against such events, we monitor the performance of model prediction. The most intuitive method to monitor a model's performance is the squared prediction error, but other more elaborate techniques have also been developed including frequency domain methods [42], two-model divergence methods [43], and OE model methods [44].

In this thesis the OE model method is used for model error monitoring. This algorithm focuses on MISO (multi-input single-output) systems, even though the simulated process is a MIMO system; however, a MIMO system can be easily constructed using several MISO systems.

Bias monitors

Sensor bias can also be a problem in control loops, as sensors are the main reference for control action. A common method for detecting sensor bias in process industries is the use of data reconciliation and gross error detection [45]. Most of data-reconciliation and gross error detection methods have been proposed for offline implementation [46]; recently, Qin and Li. [47] and Gonzalez et al. [23] developed on-line versions.

In this thesis, bias monitors for the bench scale process focus on the flow meter output versus pump speed. This type of monitor is effective for positive-displacement pumps such as those found in the bench scale process.

Process operation monitors

Process operation monitoring is a broad area of research, mainly because of the large variety of processes that can be monitored and the large number of operation phenomena that can be targeted (such as faults/breakdowns, abnormal/suboptimal operation, and violation of operating limits). Literature in this area falls under fault detection and diagnosis literature, which is reviewed in Section 1.2.1.

In this thesis, for the bench scale system, a quantitative model-driven technique is used based on the Kalman filter; here, the state is augmented in order to include two fault-related parameters (representing leaks). Under ideal conditions, the parameters have values of zero (no leak), but as leaks are introduced, the parameter values change to values significantly greater than 0.

1.3 Thesis Outline

This thesis is broken down into two major parts: *Fundamentals* and *Application*, where each of the major contributions is generally represented in both parts. The fundamentals focus on theoretical development and justification of the proposed techniques, while the application focuses on succinctly conveying all information required to apply said techniques. Because both parts are meant to be stand-alone, there may be some slight overlap between

the fundamentals and application sections, namely the parts in the fundamentals that are directly relevant to application.

A number of techniques exist in this thesis that many readers may not be familiar with, namely Bayesian diagnosis, Dempster-Shafer theory, kernel density estimation, and bootstrapping. A tutorial is provided in this thesis which covers fundamental aspects of all four techniques.

1.3.1 Problem overview and illustrative example

The main objective of this work is to diagnose the process operating mode (which contains information about the state of each process component of interest, such as sticky valves, biased sensors, inaccurate process models etc). Before diagnosing modes, we collect historical data from monitors for each mode; this historical data is used to diagnose the mode when new evidence becomes available online. Because it is assumed that corresponding modes are available with the historical data, this thesis takes a *supervised learning approach* when applying historical data.

In order to easily illustrate the challenges associated with Bayesian diagnosis, consider a problem where the modes consist of two different coins, one with a bias toward heads (probability of heads = 0.6) and one that is fair (probability of heads = 0.5). The probability estimates are obtained through historical data of coin flips. For the diagnosis problem, a coin is randomly selected and we wish to use evidence of coin flipping to determine which coin was selected. The evidence is provided by two people flipping the same coin once.

1.3.2 Previous work

The bulk of the work in this thesis follows from work presented by Pernestal [2] and by Qi [3]. In particular, Qi addressed some practical issues when applying Pernestal's work to the process industry including

Missing data

It is not uncommon in process industries that sensor records in the history are unavailable during certain time intervals. Since sensors are also used for monitoring, the corresponding monitor will also be unavailable, rendering a data point incomplete (as some elements are missing). Simply discarding incomplete data points will result in a loss of information; however, Qi and Huang [1] proposed using Bayesian methods to recover the useful information from these incomplete data points.

Using our coin-flipping example, consider the case where the evidence from the two people flipping coins is dependent. For example, after the first person flips a coin, whatever side faces up will be placed face up on the thumb on the second person. Now the historical data contains the results of both people flipping coins. In some circumstances, the result from one of the two flips will be missing. Because the coin-flips are dependent, Qi and

Huang [1] adopts Bayesian methods to use the information present to make up for the missing information.

Autodependent modes

For industrial processes, mode changes tend to be quite rare, which means that the mode at time t is highly dependent on the mode at time $t-1$. Taking this type of dependency into account can significantly increase precision of our diagnosis results, as consecutive pieces of evidence contain more information than individual pieces. This type of dependency has been addressed in Qi and Huang [4].

Returning to our coin flipping example, consider the case where after each pair of flips, there is a probability of the coin being switched. If that probability is low, the 'mode' has a strong time-wise autodependence. This means that consecutive pieces of evidence contain even more information about the mode than single pieces of evidence themselves.

Autodependent evidence

Monitor readings often use data from previous time steps in order to calculate a result. If monitor readings are not sampled slowly enough, the evidence will be autodependent. Taking the autodependence of evidence into account was addressed in Qi and Huang [5]

Autodependent evidence can also be applied to our coin flipping example. If the second coin flipper obtained heads at $t-1$, and the first coin flipper at time t placed the coin on his thumb heads-side-up (with tails being similarly treated) then results would exhibit time-wise dependence.

Sparse evidence given a mode

If a process has a large number of components, the number of possible modes will be very large. In such cases, it is quite possible for data from a particular mode to be quite sparse. Qi and Huang [48] recommended the use of bootstrapping as a method to generate additional data and get a better representation of the monitor distribution.

For the coin flipping example, consider the case where historical data for one of the coins (such as the biased one) does not have a large amount of historical data. Bootstrapping is a method that was suggested in Qi and Huang [48] to resolve this issue by simulating more coin flips by randomly drawing from previous results recorded in the historical samples.

1.3.3 Proposed work

This thesis aims to address other challenging issues that have not been previously addressed. A visual map of these solutions is given in Figure 1.2 where dark gray boxes indicate problems, medium gray boxes indicate important previously existing solutions, and white boxes indicate solutions proposed by this thesis; moreover, dotted lines indicate a combination of multiple methods. The contents of the thesis have also been accepted as a part of a book

entitled "Control System Monitoring and Diagnosis: An Evidence-Based Approach" to be published by Wiley.

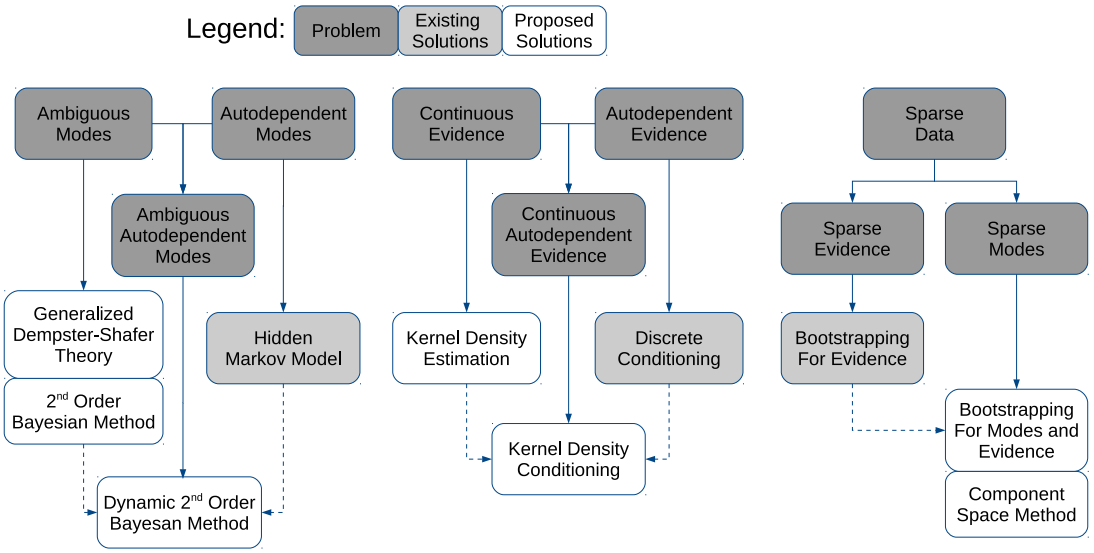


Figure 1.2: Overview of proposed solutions

Ambiguous modes from a Bayesian perspective

Qi and Huang [1] addressed the issue if some elements of historical evidence records are missing causing the evidence to be incomplete. However, just like evidence requires input from multiple monitors, the mode requires information from multiple components. If any information about the components is missing, a number of different modes will be possible, causing the mode to be *ambiguous*.

For example, if some of the historical data from coin flipping exercises contained no information on which coin was flipped (the biased or fair one) then either coin could have produced the results (heads or tails) and the corresponding mode (or coin in this case) is *ambiguous*. Since the conditioning variable is unknown, probability cannot be calculated in a straightforward manner.

Ambiguous modes from a Dempster-Shafer perspective

Demspter-Shafer theory [49] [50] has been deemed by many to be a generalization of Bayesian diagnosis that is able to handle ambiguity. However, it is shown in this work that Dempster-Shafer theory does not adequately formulate the problem of ambiguous modes in the historical data when likelihoods $p(E|M)$ are used. Some modifications are required in order to properly fit the data-driven diagnosis problem into a Dempster-Shafer framework.

Using continuous evidence

Previously, it was assumed that information used by the diagnosis method was discrete (our coin-flipping exercise yields discrete evidence). In reality however, most monitors yield an output that is continuous (for example, a monitor that monitors changes in compressor pressure). In order to reduce the amount of information lost through discretizing continuous values, this thesis proposes the use of *kernel density estimation* to make use of the continuous data directly.

Sparse or missing modes in the data

As the number of components in a system increase, the possible modes will increase exponentially. For systems with a large number of components, it is likely that data for a significant number of modes will be missing entirely.

For example, in our coin flipping exercise, even though we only have two coins (e.g. modes), we might not have any historical data from one of the coins. This thesis will present techniques on what one can do if certain modes of interest are absent from the historical data.

Dynamic applications

When accounting for ambiguous modes, the solution for addressing mode autodependence will be affected. Similarly, when accounting for continuous evidence, the solution for autodependent evidence will be affected. In Part I which deals with fundamentals, the solution to autodependent modes is addressed in Chapter 4 which discusses ambiguous modes. Likewise, the solution to autodependent evidence is addressed in Chapter 6. However, in Part II which deals with application, the solution to autodependent modes and evidence are dealt with in their own chapter (Chapter 11).

1.4 Published/Submitted Material

A number of papers have been submitted and published while completing the work done in this thesis, and material in several chapters have been addressed in these publications.

(1) R. Gonzalez, B. Huang, Control loop diagnosis with ambiguous historical operating modes: Part 1. A proportional parametrization approach, Journal of Process Control 23 (4) (2013) 585–597 [51], (Related material is found in chapters 4 and 8)

(2) R. Gonzalez, B. Huang, Control loop diagnosis from historical data containing ambiguous operating modes: Part 2. Information synthesis based on proportional parameterization, Journal of Process Control 23 (4) (2013) 1441–1454 [52] (Related material is found in chapters 4 and 8)

(3) R. Gonzalez, B. Huang, Data-driven diagnosis with ambiguous hypotheses in historical data: A generalized Dempter-Shafer approach, in: Procedings from the 16th International Conference on Information Fusion, 2013 [53] (Related material is found in chapters 5 and 9)

(4) R. Gonzalez, B. Huang, Control-loop diagnosis using continuous evidence through kernel density estimation, Journal of Process Control 24 (5) (2014) 640–651 [54] (Related material is found in chapters 6 and 10)

(5) R. Gonzalez, B. Huang, E. Lau, Process monitoring using kernel density estimation and Bayesian networking with an industrial case study, submitted to ISA Transactions, 2014 [29] (Related material is found in chapters 6 and 10)

Part I

Fundamentals

Chapter 2

Prerequisite Fundamentals

2.1 Introduction

The primary focus of this thesis is diagnosing the performance of process systems by means of historical data and Bayesian inference. However, there are many practical problems to be considered before such methods can be properly implemented, namely, missing historical evidence, ambiguous historical modes, sparse data (modes and or evidence). In addition, the Bayesian diagnosis method can be enhanced by taking into account dynamic properties of modes and evidence as well as estimating continuous distributions using kernel density estimation. Each of the solutions or enhancements make use of one of the following four tools:

1. Bayesian Inference
2. The EM Algorithm
3. Techniques for Ambiguous Modes (including Dempster-Shafer theory)
4. Kernel Density Estimation
5. Bootstrapping

Working knowledge of each of these tools is important in understanding the material presented in this thesis, thus we include a short tutorial for each of these tools in this chapter.

2.2 Bayesian Inference and Parameter Estimation

Bayesian inference is at the core of every solution mentioned in this thesis. Philosophically, Bayesian statistics interprets probability in a different manner than the more traditional Frequentist approach. The frequentist interpretation discusses problems in dealing with long-term frequencies of data generated from repeated independent random experiments [55]. However, it does not accommodate the intuitive notion that short-term probabilities exist and have meaning [56]. By contrast, the Bayesian view of probability asserts that

probability represents a subjective degree of belief, a view that was held prior to Venn by de Laplace [57] and even earlier by Bayes [30].

In practice, there are two main differences between the Bayesian and Frequentist approaches, which mainly addresses parameter estimation and inference.

Parameter estimation

When estimating parameters, the frequentist approach to parameter estimation assumes that the underlying parameters are not random and hence not subject to chance. By contrast, Bayesian methods assume that the underlying parameters are random and Bayesians must assign prior distributions to these parameters.

Consider an example where we flipped a coin 200 times, with 115 results being heads, and 85 results being tails, and we wanted to know the probability of 'heads'. The frequentist approach would be to simply estimate the probability parameter from the result ($\theta = 115/200 = 0.575$). There is no distribution associated with this result. The Bayesian approach would be to assume a Dirichlet distribution (which is explained later) which directly describes the distribution of the probability parameter θ. As we can see in Figure 2.1, the peak probability of this parameter is 0.575, (our 'heads' proportion from last time); furthermore, we can see that the distribution is fairly sharp due to the fact that we have performed this experiment about 200 times.

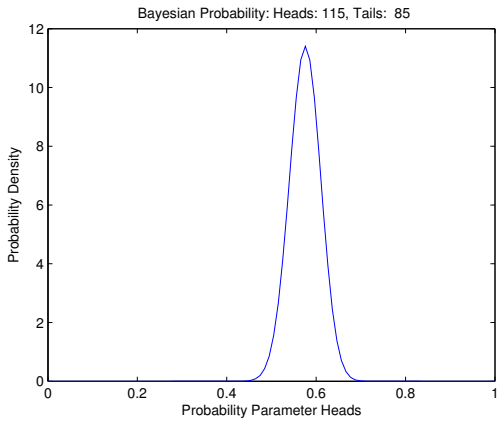


Figure 2.1: Bayesian parameter result

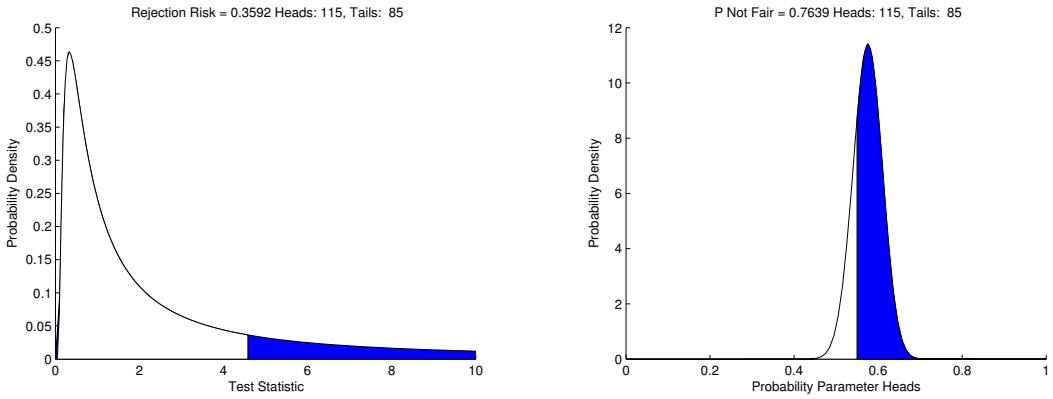
Inference about one hypothesis

When performing inference, the frequentist approach yields a probability of being false when assuming each hypothesis. The aim is to select the hypothesis that has the lowest risk of being false. The Bayesian approach will yield probabilities of each hypothesis with the aim of selecting the hypothesis with the largest probability as being true. When estimating parameters, the frequentist appraoch yields a single maximum likelihood estimate (as it is

the value with least risk of being false), while the Bayeisn approach yields a probability density function, depicting the uncertainty about the estimate.

Let us again consider the coin-flipping example, where again, the heads outcome was observed 115 times, while the tails outcome was observed 85 times and we wish to determine if the coin is fair. When performing inference, the frequentist approach first estimates the distribution of the data. Because of the central limit theorem, we can assume that the mean follows a normal distribution, enabling us to perform a T^2 test. Using the T^2 distribution, we assess the risk of being wrong if we reject the null hypothesis (that the coin is fair, or that $\mu = 0.5$). Figure 2.2(a) shows the T^2 distribution, with the shaded area being the risk of rejection (the T^2 statistic was 0.0229). From integrating the rejection region, we find that we have about a 36% risk of being wrong if we say that the coin is not fair $\mu \neq 0.5$.

The Bayesian approach to the coin problem is markedly different. First, in order to implement the Bayesian approach, we need to define what a fair coin really is. In this case, let us say that if the probability of heads θ is between 0.45 and 0.55, then the coin is fair. We use the parameter distribution to find out how much of the distribution's area lies outside of this range, which for this case (given in Figure 2.2(b)) is roughly 76%. Thus we can say that based on the observed data, there is a 24% probability that the coin's probability for heads θ lies inside our fairness interval of $[0.45, 0.55]$.



(a) Frequentist Approach

(b) Bayesian Approach

Figure 2.2: Comparison of inference methods

Inference about many hypotheses

In the coin-flipping example, the hypothesis was concerned about a continuous hypothesis, that is, the probability parameter θ. Let us consider a new example, where we have two fair coins $\theta = 0.5$ and one coin with a bias toward heads $\theta = 0.6$. Now, instead of θ being continuous (taking on an infinite number of values), it is now discrete (taking on the value of either 0.5 or 0.6). For this example, a random coin was selected and 200 trials

were performed on this coin, again with heads being observed 115 times, and tails being observed 85 times. Now we would like to determine if the selected coin is one of the fair coins ($\theta = 0.5$) or if it was the biased coin ($\theta = 0.6$).

For the frequentist approach, we calculate the T^2 statistics based on $\mu = 0.5$ and $\mu = 0.6$

$$T^2_{\theta=0.5} = \frac{(\hat{\mu} - 0.5)^2}{\hat{\sigma}^2} = \frac{(0.575 - 0.5)^2}{0.2456} = 0.0229$$
$$T^2_{\theta=0.6} = \frac{(\hat{\mu} - 0.6)^2}{\hat{\sigma}^2} = \frac{(0.575 - 0.6)^2}{0.2456} = 0.0025$$

These statistics result in rejection risks of 36% and 84% respectively. Since $\theta = 0.6$ has the highest rejection risk, we can say that we are more likely to be correct if we say that the biased coin was selected.

Now, in order to use the Bayesian approach, we must familiarize ourselves with Bayes' Theorem, the fundamental basis for all Bayesian methods.

$$p(H|E) = \frac{p(E|H)p(H)}{p(E)}$$

Here, H represents a hypothesis, E represents evidence and the probability terms are interpreted as follows

- $p(H)$ is the prior probabiltiy of the hypothesis
- $p(E|H)$ is the probability of the evidence given the hypothesis
- $p(E)$ is the probability of the evidence which can be expressed as

$$p(E) = \int p(E|H)p(H)\ dH$$

- $p(H|E)$ is the posterior probability

For our applications, H is a hypothetical value of θ (in this example it is discrete, in the previous examples, it was continuous).

In order to apply Bayes' Theorem, we calculate the likelihoods for $\theta = 0.5$ and $\theta = 0.6$ over the 200 data points.

$$p(E|\theta = 0.5) = 0.5^{115}0.5^{85} = 6.2230 \times 10^{-61}$$
$$p(E|\theta = 0.6) = 0.6^{115}0.4^{85} = 4.5972 \times 10^{-60}$$

We also have the prior probabilities based on the number of coins we have for each hypothesis

$$p(\theta = 0.5) = 2/3$$
$$p(\theta = 0.6) = 1/3$$

The prior probability is based on prior knowledge about the type of coins we have. The resulting Bayesian probabilities are therefore

$$p(\theta = 0.5|E) = \frac{(2/3)6.2230 \times 10^{-61}}{p(E)} = \frac{0.415 \times 10^{-60}}{p(E)} = 0.2131$$
$$p(\theta = 0.6|E) = \frac{(1/3)4.5972 \times 10^{-60}}{p(E)} = \frac{1.532 \times 10^{-60}}{p(E)} = 0.7869$$

From these results, we can say that it is most probable that the biased coin was selected. We can also see that the Bayesian approach has the advantage of being able to use prior probabilities (from the get-go, it was twice as probable to select a fair coin than a biased one, the frequentist approach does not take this into account). In addition, the Bayesian approach directly results in probabilities for each hypothesis which is a more intuitive result than the frequentist result of rejection risks.

Dynamic inference

One final comment about the difference between frequentist and Bayesian inference is that Bayesian inference can be easily implemented dynamically. Recall that in our Bayesian inference, we multiplied the evidence together for the 200 samples in one likelihood calculation step. However, one can obtain the same result by updating the prior probability using a single piece of evidence at a time. For example, let us say that the first observation is 'heads', then

$$p(\theta = 0.5|E_1) = \frac{(2/3)0.5}{p(E_1)} = \frac{1/3}{p(E_1)} = 5/8$$
$$p(\theta = 0.6|E_1) = \frac{(1/3)0.6}{p(E_1)} = \frac{1/5}{p(E_1)} = 3/8$$

Now let us say the second result was 'tails' then our previous posterior can be used as a prior for the next inference.

$$p(\theta = 0.5|E_1, E_2) = \frac{p(E_2|\theta = 0.5)p(\theta = 0.5|E_1)}{p(E_2)} = \frac{(5/8)0.5}{p(E_2)} = \frac{5/16}{p(E_2)} = 25/37$$
$$p(\theta = 0.6|E_1, E_2) = \frac{p(E_2|\theta = 0.6)p(\theta = 0.6|E_1)}{p(E_2)} = \frac{(3/8)0.4}{p(E_2)} = \frac{3/20}{p(E_2)} = 12/37$$

If this is continued for the entire 200 observations, we will obtain the same result as the case where all 200 observations were considered at once. For the Bayesian approach, our diagnosis result can be easily updated every time new evidence is made available, and the computational burden is the same for each new data point. Conversely, for the frequentist approach, a new test over the entire dataset has to be calculated every time a new data point is added. In this way, the computational burden increases with each new data point. This property makes the Bayesian approach more practical for on-line applications than the frequentist approach.

Practical considerations

In the scientific community, the Frequentist approach is generally more popular, understandably because parameter estimates and proposed hypotheses are not random, but take fixed values from nature. Furthermore, the aim of scientists is to perform carefully controlled experiments where results can be considered independent and identically distributed, conditions which are required for frequentist inference. Conversely, Bayesian inference is more popular in the artificial inteligence community, especially in areas where real-time decisions have to be made on hypotheses that can change at random; for example, diagnosisng problems in diesel engines [2]. Bayesian inference has the advantage that it can be easily implemented in machine learning and on-line diagnosis. Data from different scenarios can be collected off-line to estimate their respective distributions, then when implemented online, new evidence can be made to make decisions.

2.2.1 Tutorial on Bayesian inference

In this tutorial a system with two components is considered (as shown in Figure 2.3):

1. A valve which can be subject to stiction
2. A sensor which can be subject to bias

For the sake of simplicity, we assume that the two problems do not happen simultaneously, resulting in three operating modes for the process: *normal operation*, *valve stiction*, and *sensor bias*.

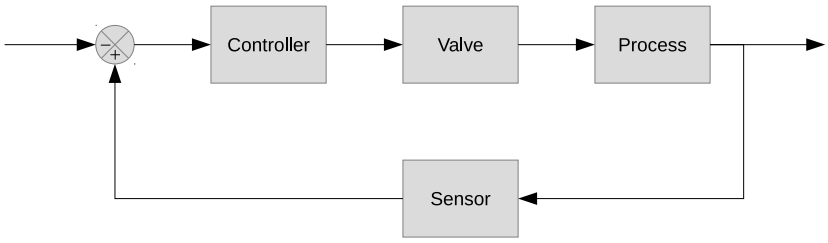


Figure 2.3: Illustrative process

Bayesian inference techniques are always applied on top of monitoring techniques. In this way, Bayesian inference was designed to piece together evidence from various sources in order to make a decision. With this in mind, we assume that monitoring algorithms are already in place to detect stiction and bias. The Bayesian technqiue is simply a layer applied above the monitors in order to make sense of monitoring input. At this point, we will assume that the monitors yield a discrete output. In the case of our example, the output for the stiction monitor is either 0 (stiction not detected) and 1 (stiction detected). Likewise,

the output for the bias monitor is either 0 (bias not detected) and 1 (bias detected). The evidence space, therefore, consists of four possible discrete values (as shown in Figure 2.4):

$$e_1 = [0, 0] \qquad e_2 = [0, 1] \qquad e_3 = [1, 0] \qquad e_4 = [1, 1]$$

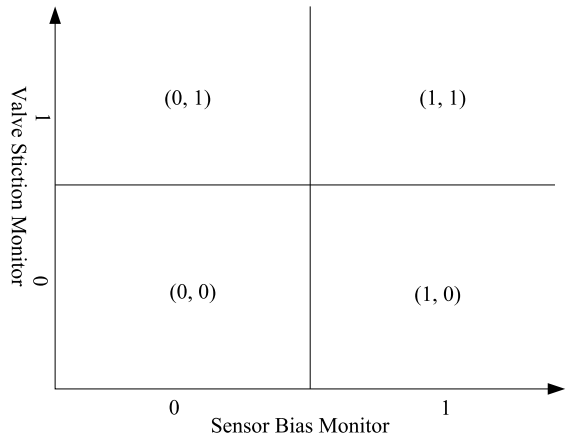


Figure 2.4: Evidence space with only prior samples

The goal of using historical evidence is to estimate the *likelihood* $p(E|M)$ for each mode. This can be combined with user-defined prior mode probabilities $p(M)$ in order to obtain a posterior

$$p(M|E) = \frac{p(E|M)p(M)}{\sum_M p(E|M)p(M)} \tag{2.1}$$

For discrete data, the likelihood $p(E|M)$ can be calculated as

$$p(E|M) = \frac{n(E, M)}{n(M)} \tag{2.2}$$

where $n(E, M)$ is the number of samples where the evidence E and mode M occur simultaneously, and $n(M)$ is the total number of samples were the mode M occurs.

The motivation for applying the Bayesian technique is alarm management. The monitors themselves are capable of generating alarms, but a problem such as stiction could also affect the sensor bias monitoring alarm. One can see that for systems with a large number components, information on the underlying problem can be obtained from the alarm pattern. Furthermore, the alarms often do not contain information about how certain the alarm is; by contrast, Bayesian techniques assign probabilities to each possible mode, allowing us to ascertain the level of uncertainty about our decision.

Bayesian methods have the added benefit of user-defined priors (denoted as $p(m)$ in Eqn (2.1)). Prior probabilities can be used to assign more weight to modes that occur more frequently. If one does not have any information about prior probabilities, a non-informative

flat prior can also be used. This can be applied to inference, but it can also be applied in estimating distribution. For example, we use non-informative prior samples for estimating likelihood distributions by assigning one data point for each discrete evidence, $a(E|m_1) = 1$, $A(m_1) = 4$, as shown in Figure 2.5. Here, all evidence possibilities are shown in a 2×2 grid (2×2 as it contains two monitors with two discrete values each), and a single sample is added to each grid sector representing a possible evidence value. By assigning a point to each grid, we state that for this mode, each possible evidence value was observed once.

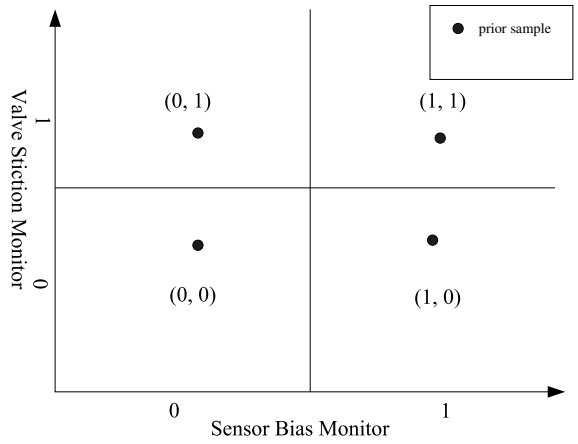


Figure 2.5: Evidence space with prior samples and historical samples

After applying the prior samples, historical samples are used to obtain the terms in Eqn (2.2)

$$n(E, M) = \text{Prior}(E, M) + \text{History}(E, M)$$
$$n(M) = \text{Prior}(M) + \text{History}(M)$$

where $\text{Prior}(E, M)$ represent the prior samples of where E and M jointly occur, and $\text{History}(E, M)$ represents the historical data samples where E and M occur. Similarly $\text{Prior}(M)$ and $\text{History}(M)$ represent the prior samples of M and the historical samples of M respectively. Table 2.1 contains results which will be used for this example

Table 2.1: Counts of historical evidence

E	Normal	Sticky Valve	Biased Sensor
$e_1 = [0, 0]$	10	0	0
$e_2 = [0, 1]$	0	1	7
$e_3 = [1, 0]$	0	8	1
$e_4 = [1, 1]$	0	1	2

When historical data and prior samples have been combined (note, prior samples placed one sample point for each possible evidence realization under each mode), the result is given in Table 2.2

Table 2.2: Counts of combined historical and prior evidence

E	Normal	Sticky Valve	Biased Sensor
$e_1 = [0, 0]$	11	1	1
$e_2 = [0, 1]$	1	2	8
$e_3 = [1, 0]$	1	9	2
$e_4 = [1, 1]$	1	2	3

As a visual example, the evidence space for the *sensor bias* mode is shown in Figure 2.6 with both prior and historical samples are shown.

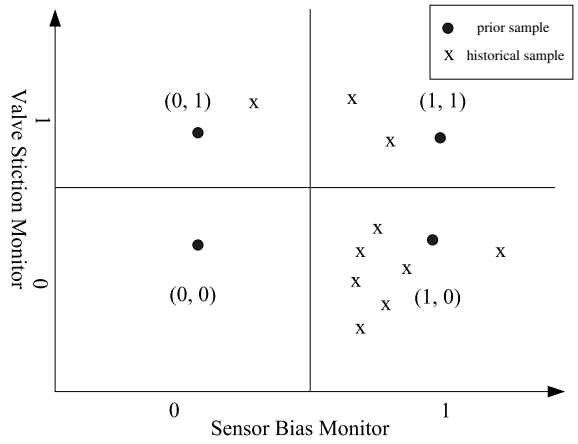


Figure 2.6: Evidence space with historical data

Likelihoods can be obtained from these samples by normalizing over the frequency of each mode. Results are shown in Table 2.3.

Table 2.3: Likelihoods of evidence

E	Normal	Sticky Valve	Biased Sensor
$e_1 = [0, 0]$	11/14	1/14	1/14
$e_2 = [0, 1]$	1/14	1/7	4/7
$e_3 = [1, 0]$	1/14	9/14	1/7
$e_4 = [1, 1]$	1/14	1/7	3/14

Before performing on-line diagnosis, priors for each mode must be assigned. For instance, if the valve has not been maintained for a considerable amount of time, then a higher prior probability can be assigned to the *Sticky Valve* mode to reflect our knowledge that the valve has a high chance of being sticky. In such a case, the prior probabilities are assigned as $p(\text{normal}) = 1/4$, $p(\text{sticky valve}) = 1/2$, and $p(\text{biased sensor}) = 1/4$.

With the estimated likelihood probabilities for current evidence E under different modes M, the likelihoods $p(E|M)$, and the user-defined prior probabilities $p(M)$, posterior probabilities of each mode $m_i \in \mathcal{M}$ can be calculated. Among these modes, the one with the largest posterior probability is selected.

As an example, given evidence $[1, 0]$ (where the stiction monitor detects a problem and the bias sensor does not) the posterior probabilities can be calculated as

$$\begin{aligned} p(\text{normal}|[1,0]) &\propto p(\text{normal}) \cdot p([1,0]|\text{normal}) \\ &= 1/4 \cdot 1/14 = 1/56 \end{aligned} \tag{2.3}$$

$$\begin{aligned} p(\text{sticky valve}|[1,0]) &\propto p(\text{sticky valve}) \cdot p([1,0]|\text{sticky valve}) \\ &= 1/2 \cdot 9/14 = 9/28 \end{aligned} \tag{2.4}$$

$$\begin{aligned} p(\text{biased sensor}|[1,0]) &\propto p(\text{biased sensor}) \cdot p([1,0]|\text{biased sensor}) \\ &= 1/4 \cdot 1/7 = 1/28 \end{aligned} \tag{2.5}$$

The mode with largest posterior probability, *Sticky Valve*, is then diagnosed as the underlying process mode. Note that these probabilities do not add up to 1 because they are not normalized by $p(E)$. If proper probabilities are desired, then normalization is required

$$p(E) = 1/56 + 9/28 + 1/28 = 3/8$$

2.2.2 Tutorial on Bayesian inference with time dependency

Mode time dependency

During on-line application, where evidence is being obtained at every Δt, unless the time intervals are exceedingly long (which would be undesirable, because shorter intervals yield more information), there will be some time dependency with the modes. In general, a mode has a probability of switching $p(M^t|M^{t-1})$ and this probability tends to be fairly small; for example, it usually takes a while for a valve to become sticky, or for an instrument to become biased, and it takes a while for these problems to be noticed and corrected. Because these switching occurrences can be rare, there tends to be strong auto-dependence within the modes, thus evidence collected over time yields more information than single pieces of evidence themselves. A visual representation of dependency is available in Figure 2.7 which resembles the well-known *Hidden Markov Model*

Mode-time dependency can be taken into account by using switching probabilities. Let

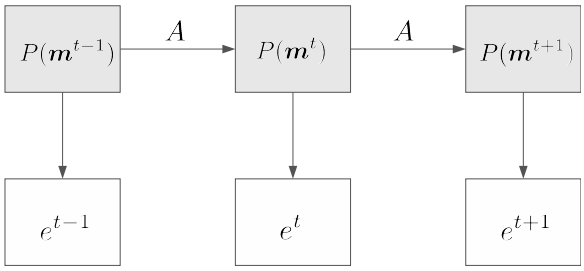


Figure 2.7: Mode dependence (Hidden Markov Model)

us consider our previous example with the following prior probabilities:

$p(m_1) = 1/2$	Normal operation
$p(m_2) = 1/4$	Sticky valve
$p(m_3) = 1/4$	Biased sensor

Now let us consider the modes having the following switching probabilities

$$A \equiv \begin{array}{lll} p(m_1^t|m_1^{t-1}) = 0.90 & p(m_1^t|m_2^{t-1}) = 0.10 & p(m_1^t|m_3^{t-1}) = 0.10 \\ p(m_2^t|m_1^{t-1}) = 0.05 & p(m_2^t|m_2^{t-1}) = 0.90 & p(m_2^t|m_3^{t-1}) = 0.00 \\ p(m_3^t|m_1^{t-1}) = 0.05 & p(m_3^t|m_2^{t-1}) = 0.00 & p(m_2^t|m_3^{t-1}) = 0.90 \end{array}$$

Let us now consider the evidence $e_1 = [0, 0]$ with the following likelihoods obtained from Table 2.3

$$p(e_1|M) = \begin{bmatrix} p(e_1|m_1) = 11/14 \\ p(e_1|m_2) = 1/14 \\ p(e_1|m_3) = 1/14 \end{bmatrix}$$

By combining the likelihood with the prior probabilities we can obtain

$$\begin{aligned} p(m_1|e_1^{t=1}) &= \frac{p(e_1|m_1)p(m_1^{t=1})}{p(e_1^{t=1})} \\ &= \frac{(11/14)(1/2)}{(11/14)(1/2) + (1/14)(1/4) + (1/14)(1/4)} \\ &= \frac{11/28}{3/7} = 11/12 \end{aligned}$$

Similarly, for the other modes, we can obtain

$$\begin{aligned} p(m_2|e_1^{t=1}) &= \frac{p(e_1|m_2)p(m_2^{t=1})}{p(e_1^{t=1})} \\ &= \frac{(1/14)(1/4)}{3/7} = 1/24 \\ p(m_3|e_1^{t=1}) &= \frac{p(e_1|m_3)p(m_3^{t=1})}{p(e_1^{t=1})} \\ &= \frac{(1/14)(1/4)}{3/7} = 1/24 \end{aligned}$$

Now let us consider the probability at $t = 2$ when e_2 is observed. Firstly, the likelihood is shown to be

$$p(e_2|M) = \begin{bmatrix} p(e_2|m_1) = 1/14 \\ p(e_2|m_2) = 1/7 \\ p(e_2|m_3) = 4/7 \end{bmatrix}$$

Now the prior probability for $t = 2$ is the posterior of $t = 1$ with switching probabilities taken into account

$$\begin{aligned} p(m_1^{t=2}|M^{t=1}) &= \sum_M p(m_1^t|M^{t-1})p(M|e_1^{t=1}) \\ &= p(m_1^t|m_1^{t-1})p(m_1|e_1^{t=1}) + p(m_1^t|m_2^{t-1})p(m_2|e_1^{t=1}) + p(m_1^t|m_3^{t-1})p(m_3|e_1^{t=1}) \\ &= 0.9 \times (11/12) + 0.1 \times (1/24) + 0.1 \times (1/24) = 5/6 \\ p(m_2^{t=2}|M^{t=1}) &= p(m_2^t|m_1^{t-1})p(m_1|e_1^{t=1}) + p(m_2^t|m_2^{t-1})p(m_2|e_1^{t=1}) + p(m_2^t|m_3^{t-1})p(m_3|e_1^{t=1}) \\ &= 0.05 \times (11/12) + 0.9 \times (1/24) + 0.0 \times (1/24) = 1/12 \\ p(m_3^{t=2}|M^{t=1}) &= p(m_3^t|m_1^{t-1})p(m_1|e_1^{t=1}) + p(m_3^t|m_2^{t-1})p(m_2|e_1^{t=1}) + p(m_3^t|m_3^{t-1})p(m_3|e_1^{t=1}) \\ &= 0.05 \times (11/12) + 0.0 \times (1/24) + 0.9 \times (1/24) = 1/12 \end{aligned}$$

These values are new priors $p(M^{t=2}|M^{t=1})$ which can be combined with the likelihoods $p(e_2^{t=2}|M)$ to obtain a new posterior $p(M^{t=2}|e_2^{t=2})$.

$$p(M^{t=2}|e_2^{t=2}) = \frac{p(e_2^{t=2}|M)p(M^{t=2}|M^{t=1})}{p(e_2^{t=2})}$$

or more generally

$$\begin{aligned} p(M^t|E^t) &= \frac{p(E^t|M)p(M^t|M^{t-1})}{p(E^t)} \\ p(m^t|M^{t-1}) &= \sum_M p(m^t|M^{t-1})p(M|E^{t-1}) \end{aligned} \tag{2.6}$$

By applying our example, we get the following numerical values for the posterior

$$\begin{aligned} p(m_1^{t=2}|e_2^{t=2}) &= \frac{p(e_2^{t=2}|m_1)p(m_1^{t=2}|M^{t=1})}{p(e_2^{t=2})} \\ &= \frac{(1/14)(5/6)}{5/42} = 1/2 \\ p(m_2^{t=2}|e_2^{t=2}) &= \frac{p(e_2^{t=2}|m_2)p(m_2^{t=2}|M^{t=1})}{p(e_2^{t=2})} \\ &= \frac{(1/7)(1/12)}{5/42} = 1/10 \\ p(m_1^{t=2}|e_2^{t=2}) &= \frac{p(e_2^{t=2}|m_1)p(m_1^{t=2}|M^{t=1})}{p(e_2^{t=2})} \\ &= \frac{(4/7)(1/12)}{5/42} = 2/5 \end{aligned}$$

Evidence time dependency

For evidence time dependency, we consider the case where E^t depends on E^{t-1}, which can happen, for example, if monitors use a window of data which overlaps with the data that monitors use at different time intervals. For the most basic case, where E^t depends on E^{t-1}, the graphical model resembles Figure 2.8.

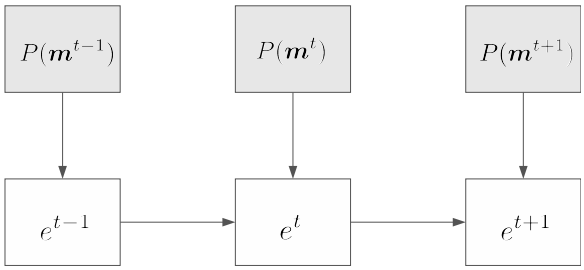


Figure 2.8: Evidence dependence

In such a case, we wish to evaluate $p(M|E^t, E^{t-1})$ as

$$p(M|E^t, E^{t-1}) = \frac{p(E^t|E^{t-1}, M)p(M)}{p(E^t)}$$

We can obtain the required likelihood expression $p(E^t|E^{t-1}, M)$ by the rule of conditioning

$$p(E^t|E^{t-1}, M) = \frac{p(M, E^t, E^{t-1})}{p(E^{t-1}, M)}$$

which yields the estimator

$$p(E^t|E^{t-1}, M^t) = \frac{n(M^t, E^t, E^{t-1})}{n(E^{t-1}, M^t)} \tag{2.7}$$

where $n(M^t, E^t, E^{t-1})$ is the number of times M^t, E^t, E^{t-1} jointly occur, and $n(E^{t-1}, M^t)$ is the number of times E^{t-1}, M^t jointly occur. This will mean that the number of evidence possibilities will be squared. For example, for evidence presented in Table 2.3, the dependent evidence solution will resemble Table 2.4.

Table 2.4: Likelihoods of dynamic evidence

E	Normal	Sticky Valve	Biased Sensor
e_1^t, e_1^{t-1}	0.5	0.01	0.01
e_1^t, e_2^{t-1}	0.07	0.07	0.01
e_1^t, e_3^{t-1}	0.07	0.01	0.07
e_1^t, e_4^{t-1}	0.07	0.01	0.01
e_2^t, e_1^{t-1}	0.07	0.07	0.01
e_2^t, e_2^{t-1}	0.01	0.5	0.01
e_2^t, e_3^{t-1}	0.01	0.07	0.07
e_2^t, e_4^{t-1}	0.01	0.06	0.01
e_3^t, e_1^{t-1}	0.07	0.01	0.7
e_3^t, e_2^{t-1}	0.01	0.07	0.7
e_3^t, e_3^{t-1}	0.01	0.01	0.5
e_3^t, e_4^{t-1}	0.01	0.01	0.7
e_4^t, e_1^{t-1}	0.06	0.01	0.01
e_4^t, e_2^{t-1}	0.01	0.07	0.01
e_4^t, e_3^{t-1}	0.01	0.01	0.06
e_4^t, e_4^{t-1}	0.01	0.01	0.01

This table of evidence can be used in the same manner as the previous table (Table 2.3). For example, if the evidence e_1^t, e_1^{t-1} was observed, the posterior would be

$$\begin{aligned}
p(m_1^t|e_1^t, e_1^{t-1}) &= p(e_1^t|e_1^{t-1}, m_1^t)p(e_1^t|e_1^{t-1}) \\
&= \frac{(0.5)(0.5)}{51/200} = 50/51 \\
p(m_2^t|e_1^t, e_1^{t-1}) &= p(e_1^t|e_1^{t-1}, m_2^t)p(e_1^t|e_1^{t-1}) \\
&= \frac{(0.01)(0.25)}{51/200} = 1/102 \\
p(m_3^t|e_1^t, e_1^{t-1}) &= p(e_1^t|e_1^{t-1}, m_3^t)p(e_1^t|e_1^{t-1}) \\
&= \frac{(0.01)(0.25)}{51/200} = 1/102
\end{aligned}$$

Dynamic evidence and modes

The dynamic evidence and dynamic modes solutions can be easily combined. The dynamic evidence solution only modifies the likelihood, while the dynamic modes solution only modifies the prior. Because the dynamic evidence solution only modifies the likelihood, we can simply substitute the dynamic evidence likelihood $p(E^t|E^{t-1}, M)$ in Eqn (2.7) for the static likelihood $p(E|M)$ which showed up in the *Hidden Markov Model* solution in Eqn (2.6). Thus, when both dynamic evidence and dynamic mode solutions are applied, one can calculate the results using

$$\begin{aligned} p(M^t|E^t, E^{t-1}) &= \frac{p(E^t|E^{t-1}, M^t)p(M^t|M^{t-1}, E^{t-1}, E^{t-2})}{p(E^t|E^{t-1})} \\ p(E^t|E^{t-1}) &= \sum_{M^t} p(E^t|E^{t-1}, M^t)p(M^t|M^{t-1}, E^{t-1}, E^{t-2}) \\ p(m_i|M^{t-1}, E^{t-1}, E^{t-2}) &= \sum_{M} p(m_i^t|M^{t-1})p(M^{t-1}|, E^{t-1}, E^{t-2}) \end{aligned} \tag{2.8}$$

As one might observe, this solution is applied in the same manner as the dynamic modes solution, but now we replace the evidence (such as Table 2.3) with dynamic evidence (such as Table 2.4). This solution solves the problem depicted in Figure 2.9.

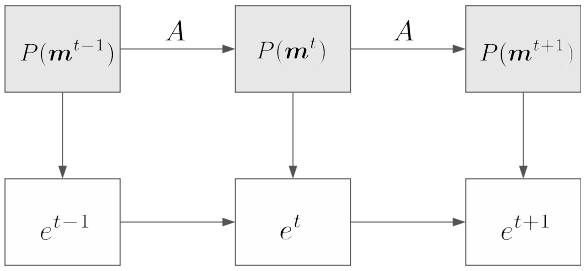


Figure 2.9: Evidence and mode dependence

2.2.3 Bayesian inference vs direct inference

Let us revisit the example where we are monitoring for sensor bias and valve stiction for the system in Figure 2.3. Now let us say that we have collected some larger samples of data with the evidence history being summarized in Table 2.5.

By looking at this table, if we observed evidence e_1, an intuitive way to evaluate the probability of the modes would be to count the occurrences of evidence e_1 for each mode ($m_1 \to 30$, $m_2 \to 7$, $m_3 \to 4$) and divide it by the total occurrence of evidence $e_1 \to 41$ so

Table 2.5: Counts of combined historical and prior evidence

E	Normal	Sticky Valve	Biased Sensor	Evidence Total
$e_1 = [0, 0]$	30	7	4	41
$e_2 = [0, 1]$	10	2	10	22
$e_3 = [1, 0]$	7	16	2	25
$e_4 = [1, 1]$	3	5	4	12
Mode Total	50	30	20	100

that

$$p(m_1|e_1) = 30/41$$
$$p(m_2|e_1) = 7/41$$
$$p(m_3|e_1) = 4/41$$

This approach is called the *direct approach* as it directly evaluates $p(M|E)$ based on historical counts. This method however, assumes that the evidence collected is representative of the mode probability. Thus, it assumes that the prior probability can be obtained from the mode totals at the bottom of Table 2.5

$$p(m_1) = 50/100 = 0.5$$
$$p(m_2) = 30/100 = 0.3$$
$$p(m_3) = 20/100 = 0.2$$

If these values were used for prior probabilities when applying the Bayesian method, the results obtained would be identical to the results from the direct method.

$$p(m_1|e_1) = \frac{p(e_1|m_1)p(m_1)}{p(e_1)} = \frac{(30/50)(50/100)}{41/100} = 30/41$$
$$p(m_2|e_1) = \frac{p(e_1|m_2)p(m_2)}{p(e_1)} = \frac{(7/30)(30/100)}{41/100} = 7/41$$
$$p(m_3|e_1) = \frac{p(e_1|m_3)p(m_3)}{p(e_1)} = \frac{(4/20)(20/100)}{41/100} = 4/41$$

Thus, as we can see, the Bayesian method is much more flexible as it allows us to select prior probabilities that are not represented in the data. This property of allowing us to use different prior probabilities means that the Bayesian method enjoys certain advantages over the direct probability method:

- **Bayesian methods allow us to collect arbitrary amounts of data for each mode.** When using Bayesian methods, priors take care of the mode probabilities so that the data does not have to.

- **Bayesian methods allows for easy on-line implementation.** When implementing solutions on-line, the prior probabilities change over time when evidence becomes available. This can be easily taken into account when using the Bayesian method, but this is much more difficult to represent when using the direct method.

2.2.4 Tutorial on Bayesian parameter estimation

In addition to Bayesian Inference, some chapters in this thesis deal with Bayesian parameter estimation. In our previous example, we have already performed some parameter estimation with respect to the probability parameters

$$p(E|M) = \theta_{E|M} = \frac{n(E,M)}{n(M)}$$

where $n(E,M)$ and $n(M)$ take into account both prior and historical samples of E and M.

However, this estimation does not yield the uncertainty behind the parameter estimate. For example, let us consider a coin flipping experiment. If we had a prior belief of one sample for heads $C = h$ and one sample for tails $C = t$, and we flipped a coin once, and observed heads, our probability of heads and tails would resemble the following

$$p(h) = \theta_h = \frac{1+1}{1+1+1+0} = \frac{2}{3}$$
$$p(t) = \theta_t = \frac{1+0}{1+1+1+0} = \frac{1}{3}$$

From our intuition however, we know that this result is not very reliable because there are so few data points. What we would like to have is to have a distribution over $\Theta = \{\theta_h, \theta_t\}$ denoted as $p(\Theta)$ that could be updated using evidence in a Bayesian manner

$$p(\Theta|E) = \frac{p(E|\Theta)p(\Theta)}{p(E)} \tag{2.9}$$

We know that $p(E|\Theta)$ is a categorical distribution, so that in our case,

$$p(h|\Theta) = \theta_h = p(h)$$
$$p(t|\Theta) = \theta_t = p(t)$$

The distribution $p(\Theta)$ is known as a *conjugate prior*, which is a distribution that can be updated by $p(E|\Theta)$. Conjugate priors are used in Bayesian parameter estimation problems in order to incorporate prior information about the parameters. These priors can be updated as information becomes available so that they reflect the uncertainty behind the updated parameter. As data becomes more available, the updated parameter distributions become narrower, depicting higher confidence in the estimates.

One of the most important properties of a conjugate prior $p(\Theta)$ to a likelihood distribution $p(E|\Theta)$ is that the posterior $p(\Theta|E)$ must be of the same family of distributions as the

prior $p(\Theta)$ after combination with the likelihood according to Eqn (2.9); this allows for easy derivation of the posterior, resulting in a computationally light updating scheme. Finding a conjugate prior is not an easy task, and not only depends on the distribution that produces E, but also on the particular parameter Θ of interest; fortunately, the conjugate priors for many likelihood distributions have been found [58] and some results are summarized in Table 2.6.

Table 2.6: List of conjugate priors

Process for generating E	**Parameter of Interest Θ**	**Conjugate Prior $p(\Theta)$**
Discrete Univariate Processes		
Bernoulli	Binary Probability (0 vs 1)	Beta
Hypergeometric	Binary Probability (0 vs 1)	Beta-binomial
Poisson	Poisson Parameter	Gamma
Discrete Multivariate Processes		
Multinomial/Categorical	Probability Parameters	Dirichlet
Multivariate Hypergeometric	Probability Parameters	Dirichlet-Multinomial
Continuous Univariate Processes		
Uniform	Uniform Probability	Pareto
Pareto	Precision Parameter	Pareto
Pareto	Shape Parameter	Gamma
Exponential	Mean	Gamma
Gamma	Rate Parameter	Gamma
Normal	Mean	Normal
Normal	Inv. Var.	Gamma
Lognormal	Normal Mean	Normal
Lognormal	Normal Inv. Var.	Gamma
Continuous Multivariate Processes		
Normal	Mean Vec.	Normal
Normal	Cov. Mat.	Wishart
Normal	Mean Vec. and Cov. Mat.	Normal-Wishart
Normal Regression	Regression Coeff.	Normal
Normal Regression	Reg. Coeff. and Precision	Normal-Gamma

For discrete probabilities, the conjugate prior is identified to be the Dirichlet Distribution [59].

$$f(\Theta|\boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_i \alpha_i\right)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i - 1} \tag{2.10}$$

$$\Theta \in [0, 1] \tag{2.11}$$

$$\sum_i \theta_i = 1 \tag{2.12}$$

where Γ is called the *gamma function*

$$\Gamma(t) = \int_0^\infty x^{t-1} e^{-x}\, dx \tag{2.13}$$

which is a generalization of the *factorial* function (as it can support continuous values) so that

$$\Gamma(k) = k! \qquad k \in \text{positive integers} \tag{2.14}$$

$$\Gamma(x+1) = x\Gamma(x) \qquad x \in \text{real numbers} \tag{2.15}$$

The parameter set $\boldsymbol{\alpha}$ can be defined as

$$\alpha_i = n(e_i)$$

so that α is simply a record of the samples. Thus, according to our coin-flipping exercise, the conjugate prior (or the function that can be updated by evidence) is given as

$$f(\Theta|\boldsymbol{\alpha}) = \frac{\Gamma\left(\alpha_h + \alpha_t\right)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \theta_h^{\alpha_h - 1} \theta_t^{\alpha_t - 1}$$

The "magic" behind the Dirichlet distribution lies within its two desirable properties

1. The expected value of θ_i is given as the intuitive fraction of data samples

$$E_{\theta_i}[f(\Theta|\boldsymbol{\alpha})] = \frac{\alpha_i}{\sum_k \alpha_k} = \frac{n(e_i)}{\sum_k n(e_k)} \tag{2.16}$$

2. Updating $f(\Theta|\boldsymbol{\alpha})$ with the observation of e_i simply adds the value of 1 to α_i

$$f(\Theta|\boldsymbol{\alpha}[k+1]) = \frac{p(E_i|\Theta) f(\Theta|\boldsymbol{\alpha}[k])}{p(E_i)} \tag{2.17}$$

$$\alpha_i[k+1] = \alpha_i[k] + 1 \tag{2.18}$$

To better understand the Dirichlet distribution, examples are given below.

Example: Expected values of a Dirichlet distribution

Let us say we have observed enough coin flips so that

$$\alpha_h = n(h) = 5$$
$$\alpha_t = n(t) = 7$$

We now wish to calculate the probability of heads, or equivalently, the expected value of θ_h.

$$\begin{aligned}
E_{\theta_h}[f(\Theta|\boldsymbol{\alpha})] &= \int_\Theta \theta_h f(\Theta|\boldsymbol{\alpha})\ d\Theta \\
&= \int_\Theta \theta_h \frac{\Gamma\left(\alpha_h+\alpha_t\right)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \theta_h^{\alpha_h-1}\theta_t^{\alpha_t-1}\ d\Theta \\
&= \frac{\Gamma\left(\alpha_h+\alpha_t\right)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \int_\Theta \theta_h \theta_h^{\alpha_h-1}\theta_t^{\alpha_t-1}\ d\Theta \\
&= \frac{\Gamma\left(\alpha_h+\alpha_t\right)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \int_\Theta \theta_h^{(\alpha_h+1)-1}\theta_t^{\alpha_t-1}\ d\Theta
\end{aligned}$$

Now it should be noted that *Dirichlet's Integral* states that

$$\int_\Theta \prod_i \theta_i^{\alpha_i-1}\ d\Theta = \frac{\prod_i \Gamma(\alpha_i)}{\Gamma\left(\sum_i \alpha_i\right)}$$

which results in

$$\begin{aligned}
E_{\theta_h}[f(\Theta|\boldsymbol{\alpha})] &= \frac{\Gamma\left(\alpha_h+\alpha_t\right)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \times \frac{\Gamma(\alpha_h+1)\Gamma(\alpha_t)}{\Gamma\left((\alpha_h+1)+\alpha_t\right)} \\
&= \frac{\Gamma\left(\alpha_h+\alpha_t\right)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \times \frac{[\alpha_h\Gamma(\alpha_h)]\Gamma(\alpha_t)}{\Gamma\left(\alpha_h+1+\alpha_t\right)} \\
&= \frac{\Gamma\left(\alpha_h+\alpha_t\right)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \times \frac{[\alpha_h\Gamma(\alpha_h)]\Gamma(\alpha_t)}{[(\alpha_h+\alpha_t)\Gamma\left(\alpha_h+\alpha_t\right)]} \\
&= \frac{1}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \times \frac{\alpha_h\Gamma(\alpha_h)\Gamma(\alpha_t)}{(\alpha_h+\alpha_t)} \\
&= \frac{\alpha_h}{(\alpha_h+\alpha_t)} = \frac{5}{5+7} = \frac{5}{12}
\end{aligned}$$

yielding the value that was intuitively expected.

Example: Updating a Dirichlet distribution with new observations

Again, let us say that historically we observed coin flips so that

$$\begin{aligned}
\alpha_h &= n(h) = 5 \\
\alpha_t &= n(t) = 7
\end{aligned}$$

Now let us say that after another coin flip, we observed heads ($E = h$) and we would like to update our results. According to Bayes' Theorem

$$f(\Theta|\boldsymbol{\alpha}, E_i) = f(\Theta|\boldsymbol{\alpha}[k+1]) = \frac{p(E_i|\Theta)f(\Theta|\boldsymbol{\alpha}[k])}{p(E_i)}$$

When applying the updating rule to our example, we have

$$\begin{aligned}
f(\Theta|\boldsymbol{\alpha}[k+1]) &= \frac{p(h|\Theta)f(\Theta|\boldsymbol{\alpha}[k])}{p(E_i)} \\
&= \frac{\theta_h f(\Theta|\boldsymbol{\alpha}[k])}{\int_\Theta \theta_h f(\Theta|\boldsymbol{\alpha}[k])\ d\Theta}
\end{aligned}$$

Now, from our previous expected value example, we have already shown that

$$\begin{aligned}\int_{\Theta} \theta_h f(\Theta|\boldsymbol{\alpha}[k])\ d\Theta &= E_{\theta_h} f(\Theta|\boldsymbol{\alpha}[k]) \\ &= \frac{\alpha_h}{\alpha_h + \alpha_t}\end{aligned}$$

Thus

$$\begin{aligned}f(\Theta|\boldsymbol{\alpha}[k+1]) &= \frac{p(h|\Theta) f(\Theta|\boldsymbol{\alpha}[k])}{p(E_i)} \\ &= \frac{\theta_h f(\Theta|\boldsymbol{\alpha}[k])}{\frac{\alpha_h}{\alpha_h+\alpha_t}} \\ &= \frac{\alpha_h + \alpha_t}{\alpha_h} f(\Theta|\boldsymbol{\alpha}[k])\theta_h \\ &= \frac{\alpha_h + \alpha_t}{\alpha_h} \theta_h \frac{\Gamma(\alpha_h + \alpha_t)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \theta_h^{\alpha_h - 1} \theta_t^{\alpha_t - 1} \\ &= \frac{\alpha_h + \alpha_t}{\alpha_h} \frac{\Gamma(\alpha_h + \alpha_t)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \theta_h^{(\alpha_h+1)-1} \theta_t^{\alpha_t - 1}\end{aligned}$$

Recalling in Eqn (2.15) that $\Gamma(x+1) = x\Gamma(x)$,

$$\begin{aligned}f(\Theta|\boldsymbol{\alpha}[k+1]) &= \frac{\Gamma(\alpha_h + \alpha_t + 1)}{\Gamma(\alpha_h + 1)\Gamma(\alpha_t)} \theta_h^{(\alpha_h+1)-1} \theta_t^{\alpha_t - 1} \\ &= \frac{\Gamma((\alpha_h + 1) + \alpha_t)}{\Gamma(\alpha_h + 1)\Gamma(\alpha_t)} \theta_h^{(\alpha_h+1)-1} \theta_t^{\alpha_t - 1}\end{aligned}$$

From this, we can see that the posterior probability $f(\Theta|\boldsymbol{\alpha}[k+1])$ has $(\alpha_h + 1)$ in every place that (α_h) occurred in $f(\Theta|\boldsymbol{\alpha}[k])$. Therefore,

$$\alpha_h[k+1] = \alpha_h + 1 = 5 + 1 = 6$$

when a new outcome of "heads" $(E = h)$ is observed.

2.3 The EM Algorithm

The EM algorithm is a technique pioneered by Dempster et al. [60]; variations of the EM Algorithm technique previously proposed, but [60] was the first to present it in a general manner with rigorous proof of convergence. The principal reason for implementing the EM algorithm is to learn relationships and distributions when data for estimation is incomplete.

Dempster et al. [60] presented the EM algorithm in three different forms having increased generality:

1. Exponential family distributions with closed-form maximum-likelihood solutions
2. Exponential family distributions without closed-form maximum-likelihood solutions
3. General distributions

However, since all solutions can be obtained from the general solution, we set our focus on the most general solution.

Tutorial problem

For the EM Algorithm tutorial, we revisit the simple control loop example as presented in Figure 2.3. Again, we have two monitors, one which monitors instrument bias, while another monitors valve stiction. Consider a mode *Biased Sensor* where we have some missing values from certain monitors as shown in Table 2.7. When a monitor's value is missing, we denote it as $\times$

Table 2.7: *Biased sensor* mode

Evidence	Frequency
$[0, 0]$	5
$[0, 1]$	12
$[1, 0]$	4
$[1, 1]$	6
$[0, \times]$	8
$[1, \times]$	2
$[\times, 0]$	3
$[\times, 1]$	10
Total	50

We would like to estimate $p(E|m_3)$ when pieces of data are missing from the evidence.

2.3.1 Tutorial: Solution for general distributions

The most general type of solution does not rely on any notion of exponential families. However, becuase it is the most general, the other two solutions can be derived as a special case of this solution. Thus, the general solution is most often used as a starting point despite its complexity.

1. **Expectation:** This step involves the construction of the Q function

$$Q(\Phi|\Phi^{[k]}) = E\left[\log f(Z|\Phi)|y, \Phi^{[k]}\right] = \int_Z \log(f(Z|\Phi))p(Z|y, \Phi^{[k]})\ dZ$$

   where Z represents the unobserved part of the dataset (notation is upper case due to its random nature), and y represents the observed part of the dataset (notation is lower-case due to its non-random nature). Furthermore, Φ is the current (variable) parameter set, and $\Phi^{[k]}$ is the previously obtained (constant) parameter set obtained at iteration k.

2. **Maximization:** This step involves numerical maximization of the Q function over the variable parameter set Φ

$$\Phi^{[k+1]} = \arg\max_{\Phi} E(\log f(Z|\Phi)|y, \Phi^{[k]}))$$

where $\Phi^{[k+1]}$ is the estimate of parameters Φ after iteration k

In order to apply this solution to our tutorial problem, we must first take note of the following notation.

- $\mathcal{D}_c$ is a matrix of complete data entries
- $\mathcal{D}_{ic}$ is a matrix of incomplete data entries (containing missing "×" elements)
- $\boldsymbol{Z}$ is a random vector consisting of the missing (or ×) elements within $\mathcal{D}_{ic}$
- $\boldsymbol{z}$ is a realization of $\boldsymbol{Z}$
- $\boldsymbol{y}$ is a vector which represents all of the observed elements within $\mathcal{D}_{ic}$

When referring to the historical data in our sample problem (shown in Table 2.7) $\mathcal{D}_c$ represents the data points without any missing entries (e.g. data summarized in the first four rows), while $\mathcal{D}_{ic}$ represents the data points that have missing entries (e.g. data summarized in the last four rows). In addition, $\boldsymbol{Z}$ represents all occurrences of missing elements (×) in $\mathcal{D}_{ic}$, while $\boldsymbol{y}$ represents all occurrences of observed elements (not ×) in $\mathcal{D}_{ic}$.

Then, when implementing the EM Algorithm in our example problem, one goes through the following steps:

1. **Initialization:** As an initialization point, we calculate the $\Theta^{[0]}$ parameters without any missing data

$$\theta_{[0,0]} = 5/27$$
$$\theta_{[0,1]} = 12/27$$
$$\theta_{[1,0]} = 4/27$$
$$\theta_{[1,1]} = 6/27$$

2. **Expectation:** This is the step that involves the bulk of the work. We must take the Q function, which is initially expressed as

$$Q(\Theta|\Theta^{[k]}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{z}|\boldsymbol{y}, D_c, \Theta^p) \log p(\boldsymbol{y}, \boldsymbol{z}, D_c|\Theta)$$

and derive a usable expression from it. The reason this current expression is hard to implement is that there are $2^{n(\boldsymbol{z})}$ possible realizations of $\boldsymbol{Z}$, thus, summation becomes an infeasible exercise. The first simplification can be made by assuming time-independence of the data. In this way

$$\mathcal{D}_c \perp \mathcal{D}_{ic}$$
$$\mathcal{D}_c \perp \{\boldsymbol{z}, \boldsymbol{y}\}$$

This essentially removes $\mathcal{D}_c$ from the conditional probability expression, allowing us to express it as a separate independent term

$$Q(\Theta|\Theta^{[k]}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{z}|\boldsymbol{y}, \Theta^{[k]}) \left[\log p(\boldsymbol{y}, \boldsymbol{z}|\Theta) + \log p(\mathcal{D}_c|\Theta)\right]$$

Now, we have already established that $\mathcal{D}_c$ is independent of $\boldsymbol{Z}$ which is the summation term. Because

$$\sum_{\boldsymbol{Z}} p(\boldsymbol{z}|\boldsymbol{y}, \Theta^{[k]}) = 1$$

we can separate out the $\log p(\mathcal{D}_c|\Theta)$ from the summation

$$Q(\Theta|\Theta^{[k]}) = \log p(\mathcal{D}_c|\Theta) + \sum_{\boldsymbol{Z}} p(\boldsymbol{z}|\boldsymbol{y}, \Theta^{[k]}) \log p(\boldsymbol{y}, \boldsymbol{z}|\Theta)$$

Now, because of time-independence, the summation over $\boldsymbol{Z}$ can be broken down into the elements in $t_1, t_2, \ldots, t_{n(\mathcal{D}_ic)}$. In terms of $\boldsymbol{Z}$ this means that the realizations are broken down into

$$\boldsymbol{Z} = \left\{Z^1, Z^2, \ldots Z^{n(\mathcal{D}_{ic})}\right\} \tag{2.19}$$

where Z^t is the set of realizations of all $\times$ values in the t^{th} incomplete data sample (in $\mathcal{D}_{ic}$). For example,

- If the first element of $\mathcal{D}_{ic}$ was $[0, \times]$ then $Z^1 = \{0, 1\}$.
- If the second element of $\mathcal{D}_{ic}$ was $[\times, 1]$ then $Z^2 = \{0, 1\}$.
- If the third element of $\mathcal{D}_{ic}$ was $[\times, \times]$ then $Z^3 = \{[0, 0], [0, 1], [1, 0], [1, 1]\}$ or it could also be simply mapped to four possible values $Z^3 = \{0, 1, 2, 3\}$.

Note that in being consistent with previous notation, z^t is one of the possible realizations of Z^t so that

$$\sum_i p(Z^t = z_i^t) = 1$$

When time-independence is taken into account, we can assert that

$$p(\boldsymbol{Z}|\boldsymbol{y}, \Theta^{[k]}) = \prod_{t=1}^{n(\mathcal{D}_{ic})} p(Z^t|y^t, \Theta^{[k]})$$

$$\log p(\boldsymbol{Z}, \boldsymbol{y}|\Theta^{[k]}) = \sum_{t=1}^{n(\mathcal{D}_{ic})} \log p(Z^t, y^t|\Theta^{[k]})$$

This results in a summation that is broken down as follows:

$$Q(\Theta|\Theta^{[k]}) = \log p(\mathcal{D}_c|\Theta) + \sum_{Z^1} \cdots \sum_{Z^{n(\mathcal{D}_{ic})}} \left[\prod_{t=1}^{n(\mathcal{D}_{ic})} p(z^t|y^t, \Theta)\right] \left[\sum_{t=1}^{n(\mathcal{D}_{ic})} \log p(z^t, y^t|\Theta^{[k]})\right]$$

We can simplify this expression by noting that each term of $\log p(z^t, y^t|\Theta^{[k]})$ is front-multiplied by all values of $p(z^t|y^t, \Theta)$; however, the first term $\log p(z^1, y^1|\Theta^{[k]})$ is independent of all terms in $p(z^t|y^t, \Theta)$ except $p(z^1|y^1, \Theta)$. And because the summation of all realizations $p(z^t|y^t, \Theta)$ over Z^t is equal to unity, the summations signs all cancel out, except for the one occurring at $t = 1$

$$\sum_{Z^1} \cdots \sum_{Z^{n(\mathcal{D}_{ic})}} \left[\prod_{t=1}^{n(\mathcal{D}_{ic})} p(z^t|y^t, \Theta) \right] \log p(z^1, y^t|\Theta^{[k]}) = \sum_{Z^1} p(z^1|y^1, \Theta) \log p(z^1, y^1|\Theta^{[k]})$$

This property can be generalized to yield

$$\sum_{Z^1} \cdots \sum_{Z^{n(\mathcal{D}_{ic})}} \left[\prod_{t=1}^{n(\mathcal{D}_{ic})} p(z^t|y^t, \Theta) \right] \log p(z^i, y^i|\Theta^{[k]}) = \sum_{z^i} p(z^i|y^i, \Theta) \log p(z^i, y^i|\Theta^{[k]})$$

When applied to our Q function, the simplification yields the following result:

$$Q(\Theta|\Theta^{[k]}) = \log p(\mathcal{D}_c|\Theta) + \sum_{t=1}^{n(\mathcal{D}_{ic})} \sum_{Z^t} p(z^t|y^t, \Theta) \log p(z^t, y^t|\Theta^{[k]})$$

This is a more workable solution. For example if each incomplete data entry $\mathcal{D}_{ic}$ has only one element missing, instead of assessing the probability of $2^{n(\mathcal{D}_{ic})}$ realizations of $\boldsymbol{Z}$ and summing them up, we only have to go through 2 realizations for every Z^t and summing up the results. Thus the computational burden of evaluating the $Q(\Theta|\Theta^{[k]})$ function only increases linearly instead of exponentially.

3. **Maximization:** The maximization step could be performed by numerical optimization

$$\Theta^{[k+1]} = \arg\max_{\Theta} \left[\log p(\mathcal{D}_c|\Theta) + \sum_{t=1}^{n(\mathcal{D}_{ic})} \sum_{Z^t} p(z^t|y^t, \Theta) \log p(z^t, y^t|\Theta^{[k]}) \right]$$

However, the solution to this can be analytically obtained. In order to obtain a result that relates more directly to our original problem, we first note that

$$\begin{aligned} \log p(\mathcal{D}_c|\Theta) &= \sum_{t=1}^{n(\mathcal{D}_c)} \log p(e_c^t|\Theta) \\ &= n_c[0,0] \log(\theta_{[0,0]}) + n_c[0,1] \log(\theta_{[0,1]}) + \\ &\quad n_c[1,0] \log(\theta_{[1,0]}) + n_c[1,1] \log(\theta_{[1,1]}) \end{aligned}$$

Now recall that z^t, y^t will correspond to some e^t_{ic}. For example, let us say that $[0, \times]$ is the first element of D_{ic}. Then

$$\{z^1 = 0\} \mapsto \{[0, \times] = [0, 0] = e_1\}$$
$$\{z^1 = 1\} \mapsto \{[0, \times] = [0, 1] = e_2\}$$

thus

$$p(z^1 = 0, y^1 | \Theta^{[k]}) = p(e^1_{ic} = [0, 0] | \Theta^{[k]})$$
$$p(z^1 = 1, y^1 | \Theta^{[k]}) = p(e^1_{ic} = [0, 1] | \Theta^{[k]})$$

In addition, because of the rule of conditioning

$$p(z^t | y^t, \Theta) = \frac{p(z^t, y^t | \Theta)}{p(y^t | \Theta)}$$

Now $p(y^t|\Theta)$ is the probability that y_t takes this value, or equivalently it is a normalization constant to ensure that all probabilities of $p(z^t|y^t, \Theta)$ sum to 1 (with respect to all possible values of z^t). Thus according to our example, we can say that

$$\begin{aligned} p(Z^1 = 0 | y^1, \Theta^{[k]}) &= \frac{p(E^1_{ic} = [0, 0] | \Theta^{[k]})}{p(E^1_{ic} = [0, 0] | \Theta^{[k]}) + p(E^1_{ic} = [0, 1] | \Theta^{[k]})} \\ &= \frac{\theta_{[0,0]}}{\theta_{[0,0]} + \theta_{[0,1]}} \\ p(Z^1 = 1 | y^1, \Theta^{[k]}) &= \frac{p(E^1_{ic} = [0, 1] | \Theta^{[k]})}{p(E^1_{ic} = [0, 0] | \Theta^{[k]}) + p(E^1_{ic} = [0, 1] | \Theta^{[k]})} \\ &= \frac{\theta_{[0,1]}}{\theta_{[0,0]} + \theta_{[0,1]}} \end{aligned}$$

so that in general,

$$p(z^t | y^t, \Theta^{[k]}) = \frac{\theta_{E^t_{ic}}}{\sum\limits_{e \subset E^t_{ic}} \theta_e}$$

where $\sum\limits_{e \subset E^t_{ic}} \theta_e$ sums the θ parameters associated with all possible evidence realizations in E^t_{ic}. For example,

$$\sum_{e \subset [\times, 0]} \theta_e = \theta_{[1,0]} + \theta_{[0,0]}$$

Now $Q(\Theta|\Theta^{[k]})$ can be rewritten as

$$
\begin{aligned}
Q(\Theta|\Theta^{[k]}) = \\
& n_c[0,0]\log(\theta_{[0,0]}) + n_c[0,1]\log(\theta_{[0,1]}) \\
& + n_c[1,0]\log(\theta_{[1,0]}) + n_c[1,1]\log(\theta_{[1,1]}) \\
& + n_{ic}[0,\times]\left[\log(\theta_{[0,0]})\frac{\theta_{[0,0]}}{\theta_{[0,0]}+\theta_{[0,1]}} + \log(\theta_{[0,1]})\frac{\theta_{[0,1]}}{\theta_{[0,0]}+\theta_{[0,1]}}\right] \\
& + n_{ic}[1,\times]\left[\log(\theta_{[1,0]})\frac{\theta_{[1,0]}}{\theta_{[1,0]}+\theta_{[1,1]}} + \log(\theta_{[1,1]})\frac{\theta_{[1,1]}}{\theta_{[1,0]}+\theta_{[1,1]}}\right] \\
& + n_{ic}[\times,0]\left[\log(\theta_{[1,0]})\frac{\theta_{[1,0]}}{\theta_{[1,0]}+\theta_{[0,0]}} + \log(\theta_{[0,0]})\frac{\theta_{[0,0]}}{\theta_{[1,0]}+\theta_{[0,0]}}\right] \\
& + n_{ic}[\times,1]\left[\log(\theta_{[1,1]})\frac{\theta_{[1,1]}}{\theta_{[1,1]}+\theta_{[0,1]}} + \log(\theta_{[0,1]})\frac{\theta_{[0,1]}}{\theta_{[1,1]}+\theta_{[0,1]}}\right]
\end{aligned}
$$

Which can be rearranged as

$$
\begin{aligned}
Q(\Theta|\Theta^{[k]}) = & \log\theta_{[0,0]}\left[n_c[0,0] + n_{ic}[\times,0]\frac{\theta_{[0,0]}}{\theta_{[0,0]}+\theta_{[1,0]}} + n_{ic}[0,\times]\frac{\theta_{[0,0]}}{\theta_{[0,0]}+\theta_{[0,1]}}\right] \\
& + \log\theta_{[0,1]}\left[n_c[0,1] + n_{ic}[\times,1]\frac{\theta_{[0,1]}}{\theta_{[0,1]}+\theta_{[1,1]}} + n_{ic}[0,\times]\frac{\theta_{[0,1]}}{\theta_{[0,0]}+\theta_{[0,1]}}\right] \\
& + \log\theta_{[1,0]}\left[n_c[1,0] + n_{ic}[\times,0]\frac{\theta_{[1,0]}}{\theta_{[0,0]}+\theta_{[1,0]}} + n_{ic}[1,\times]\frac{\theta_{[1,0]}}{\theta_{[1,0]}+\theta_{[1,1]}}\right] \\
& + \log\theta_{[1,1]}\left[n_c[1,1] + n_{ic}[\times,1]\frac{\theta_{[1,1]}}{\theta_{[0,1]}+\theta_{[1,1]}} + n_{ic}[1,\times]\frac{\theta_{[1,1]}}{\theta_{[1,0]}+\theta_{[1,1]}}\right]
\end{aligned}
$$

By applying numbers in Table 2.7 , $Q(\Theta|\Theta^{[k]})$ is expressed as

$$
\begin{aligned}
Q(\Theta|\Theta^{[k]}) = & \log\theta_{[0,0]}\left[5 + 3\frac{5/27}{5/27+4/27} + 8\frac{5/27}{5/27+12/27}\right] \\
& + \log\theta_{[0,1]}\left[12 + 10\frac{12/27}{12/27+6/27} + 8\frac{12/27}{12/27+5/27}\right] \\
& + \log\theta_{[1,0]}\left[4 + 3\frac{4/27}{4/27+5/27} + 2\frac{4/27}{4/27+6/27}\right] \\
& + \log\theta_{[1,1]}\left[6 + 10\frac{6/27}{6/27+12/27} + 2\frac{6/27}{6/27+4/27}\right]
\end{aligned}
$$

resulting in the following expression

$$
Q(\Theta|\Theta^{[k]}) = 9.0196\log\theta_{[0,0]} + 24.314\log\theta_{[0,1]} + 6.133\log\theta_{[1,0]} + 10.533\log\theta_{[1,1]}
$$

When maximized over all values in Θ, in light of the constraint

$$
\theta_{[0,0]} + \theta_{[0,1]} + \theta_{[1,0]} + \theta_{[1,1]} = 1 \tag{2.20}
$$

the result is

$$\theta^{[1]}_{[0,0]} = 9.0196/50$$
$$\theta^{[1]}_{[0,1]} = 24.314/50$$
$$\theta^{[1]}_{[1,0]} = 6.133/50$$
$$\theta^{[1]}_{[1,1]} = 10.533/50$$

which is an intuitive result. The *expectation* and *maximization* procedures can be repeated until the parameters in Θ converge.

2.4 Techniques for Ambiguous Modes

In Section 2.3 we discussed the problem of missing evidence and how the EM algorithm could be used to infer likelihoods even if some of the historical evidence is incomplete. In this section, we can discuss how to infer likelihoods if some of the modes are incomplete.

If we return to our example system presented in Figure 2.3, where we have a sensor that could be biased, and a valve that could become sticky. Let us say that in this case, it is possible for more than one problem to exist at a given time. In this way, there are four modes m_1, m_2, m_3, m_4 as shown in Table 2.8.

Table 2.8: Modes and their corresponding labels

Mode Label	Meaning	Binary Label
m_1	no bias, no stiction	$[0, 0]$
m_2	no bias, stiction	$[0, 1]$
m_3	bias, no stiction	$[1, 0]$
m_4	bias, stiction	$[1, 1]$

As in the case of evidence, it is possible for information about the mode to be missing in the history. In such a case, the mode is ambiguous. For example, let us say that we are sure that there is no bias, but we are unsure as to whether stiction exists or not. The binary label for such a mode would be $[0, \times]$, which would indicate that the modes m_1 and m_2 were possible (resulting in a mode label $\{m_1, m_2\}$). Let us now assume that additional modes were seen in the data according to Table 2.9.

Table 2.9: Ambiguous modes and their corresponding labels

Mode Label	Meaning	Binary Label
$\{m_1, m_2\}$	no bias, stiction uncertain	$[0, \times]$
$\{m_3, m_4\}$	bias, stiction uncertain	$[1, \times]$
$\{m_1, m_3\}$	bias uncertain, no stiction	$[\times, 0]$
$\{m_2, m_4\}$	bias uncertain, stiction uncertain	$[\times, 1]$

For the tutorials to follow, let us consider some historical data as seen in Table 2.10.

Table 2.10: Historical data for all modes

	Modes								
Evidence	$[0,0]$	$[0,1]$	$[1,0]$	$[1,1]$	$[0,\times]$	$[1,\times]$	$[\times,0]$	$[\times,1]$	**Total**
$e_1, [0,0]$	11	1	1	1	6	2	6	1	29
$e_2, [0,1]$	1	2	8	2	1	2	4	2	22
$e_3, [1,0]$	1	8	2	2	5	4	2	5	29
$e_4, [1,1]$	1	2	3	9	2	6	2	6	31
Total	14	14	14	14	14	14	14	14	112

From Table 2.10, we can see that fourteen data points were collected from every mode (including the ambiguous ones). If we assume that the mode frequency in the data represents the true mode frequency (so that all modes have an equal chance of happening as presented in the data) we could use the EM algorithm [60] to solve this problem. However, because we are using Bayesian diagnosis methods

$$p(M|E) = \frac{p(E|M)p(M)}{p(E)}$$

the term $p(M)$ indicates that we have a prior probability of modes that is *not obtained from historical data*. The Bayesian evaluation of $p(M|E)$ is convenient because it allows us more freedom in terms of how we select data (for example, if mode 1 occurs 95% of the time, we do not need to ensure that 95% of the historical data comes from mode 1). Now, the EM algorithm in this case, would use the data to find the mode probability $p(M)$; however, if we assume that we cannot use the historical data to estimate the mode frequency, as is done in Bayesian diagnosis, the EM algorithm should not be applied as it would attempt to estimate the ambiguous mode statistics (defined as Θ) using historical data.

Instead, we can we can use unknown parameters Θ to express the likelihood based on different outcomes of the ambiguous modes. These parameters can be used to express a probability ranges in the diagnosis. This type of approach (using ambiguity to express probability ranges) has been introduced in the topic of Dempster-Shafer theory but as we

will see in Section 2.4.2 the expression of $p(E|\Theta)$ is somewhat complicated. Furthermore, we will see that in Section 2.4.3 the problem of *Bayesian probability* (with ambiguous modes) cannot be well-represented by the formulation given by Shafer [50], although the problem of *direct probability* can be represented quite well by this same formulation.

2.4.1 Tutorial on Θ parameters in the presence of ambiguous modes

Previously, we used Θ to express the probability $\theta_i = p(e_i|M)$. In the case of ambiguous modes, Θ is also used to express probability, but now we are concerned about the probability of the mode

$$p(m_i) = \theta\{m_i\}$$

More specifically however, we are focused on the probability of an unambiguous mode m_i, given a historical mode $\boldsymbol{m}_k$ which could potentially be ambiguous (a boldface $\boldsymbol{m}$ here, indicates that the mode can be ambiguous).

$$\theta\{\tfrac{m_i}{\boldsymbol{m}_k}\} \equiv p(m_i|\boldsymbol{m}_k)$$

For example, let us consider the historical mode $\{m_1, m_2\}$ which is ambiguous (refer to Table 2.9). The parameter θ for mode m_1 given this ambiguous mode is

$$\theta\{\tfrac{m_1}{m_1,m_2}\} = p(m_1|\{m_1, m_2\})$$

In other words, $\theta\{\frac{m_1}{m_1,m_2}\}$ is the amount of data in $\{m_1, m_2\}$ that actually belongs to m_1. In general, the values of θ are unknown parameters except in the following cases

(1)	$\theta\{\tfrac{m_i}{\boldsymbol{m}_k}\} = p(m_i\|\boldsymbol{m}_k) = 0$	$m_i \nsubseteq \boldsymbol{m}_k$
e.g.	$\theta\{\tfrac{m_3}{m_1,m_2}\} = p(m_3\|\{m_1, m_2\}) = 0$	
(2)	$\theta\{\tfrac{m_i}{\boldsymbol{m}_k}\} = p(m_i\|\boldsymbol{m}_k) = 1$	$m_i = \boldsymbol{m}_k$
e.g.	$\theta\{\tfrac{m_1}{m_1}\} = p(m_1\|m_1) = 1$	

One can see that in these special cases, logic forces the probability to be 1 or 0, hence, these cases are *logically forced*.

2.4.2 Tutorial on probabilities using Θ parameters

Now that an interpretation on Θ has been given, we can proceed to express probabilities of unambiguous modes given the data (which includes ambiguous modes).

Direct probability

Let us first assume that we are not using Bayes' theorem, but that we are using the data $p(M|E)$ to directly evaluate the probability. In order to do this properly, one must assume that the mode frequencies in the data are representative of the true mode frequency (or that the number of times the mode happens in the data is proportional to its probability).

Let us say that from our system in Figure 2.3, we observe the evidence $E = [0, 0]$ and we wish to evaluate the probability of mode $m_1 = [0, 0]$. Now in addition to the eleven data points we have observed for this evidence under $m_1 = [0, 0]$, we must also consider the data points under $\boldsymbol{M} = [0, \times], [\times, 0]$ or equivalently $\boldsymbol{M} = \{m_1, m_2\}, \{m_1, m_3\}$ wherein some of those data points could also belong to $m_1 = [0, 0]$. The amount of data in $\boldsymbol{M} = \{m_1, m_2\}, \{m_1, m_3\}$ that does belong to m_1 is given by the parameters $\theta\{\frac{m_1}{m_1,m_2}\}, \theta\{\frac{m_1}{m_1,m_3}\}$ which are unknown. The probability of m_1 is then expressed as

$$p(m_1|e_1, \Theta) = \frac{\left[n(m_1, e_1) + n(\{m_1, m_2\}, e_1)\theta\{\frac{m_1}{m_1,m_2}\} + n(\{m_1, m_3\}, e_1)\theta\{\frac{m_1}{m_1,m_3}\}\right]}{n(e_1)}$$

Now $n(m_1, e_1), n(\{m_1, m_2\}, e_1)$, and $n(\{m_1, m_3\}, e_1)$ are the number of observations of the modes $[0, 0], [0, \times]$, and $[\times, 0]$ when $e = [1, 0]$. These values can be obtained from Table 2.10 as 11, 6, and 6 respectively. Similarly $n(e_1)$ is likewise found to be 29 so that

$$p(m_1|e_1, \Theta) = \frac{\left[11 + 6\ \theta\{\frac{m_1}{m_1,m_2}\} + 6\ \theta\{\frac{m_1}{m_1,m_3}\}\right]}{29}$$

In general, the probability of $p(M|E, \Theta)$ is given as

$$\begin{aligned} P(M|E, \Theta) &= \frac{1}{n(E)} \sum_{\boldsymbol{m}_k \supseteq M} \theta\{\tfrac{M}{\boldsymbol{m}_k}\} n(\boldsymbol{m}_k, E) \\ &= \sum_{\boldsymbol{m}_k \supseteq M} \theta\{\tfrac{M}{\boldsymbol{m}_k}\} S(\boldsymbol{m}_k|E) \end{aligned} \tag{2.21}$$

where the summation condition $\boldsymbol{m}_k \supseteq M$ indicates that we search through all historical modes and sum over the terms $\boldsymbol{m}_k$ when it contains or is equal to M. The term $S(\boldsymbol{m}_k|E)$ was coined by Shafer [50] as *support*, or in later terminology, the *Basic Belief Assignment* (BBA). The support function is functionally the same as probability, but makes allowances for ambiguous modes; for example,

$$S(\boldsymbol{m}_k|E) = \frac{n(\boldsymbol{m}_k, E)}{n(E)}$$

Bayesian probability

When using the Bayesian technique for diagnosis, we combine the prior probability $p(M)$ with the likelihood $p(E|M)$, where the likelihood is calculated from historical data

$$p(E|M) = \frac{n(E, M)}{n(M)}$$

When ambiguous modes are in the history, we have to consider not only the data points from mode M but also the data points from ambiguous modes that could also belong to M. For example, if we wish to consider the likelihood of $E = e_1 = [0, 0]$ given $M = m_1$, the likelihood $p(e_1|m_1, \Theta)$ is expressed as

$$p(e_1|m_1,\Theta) = \frac{n(e_1,m_1) + \theta\{\frac{m_1}{m_1,m_2}\}n(e_1,\{m_1,m_2\}) + \theta\{\frac{m_1}{m_1,m_3}\}n(e_1,\{m_1,m_3\})}{n(m_1) + \theta\{\frac{m_1}{m_1,m_2}\}n(\{m_1,m_2\}) + \theta\{\frac{m_1}{m_1,m_3}\}n(\{m_1,m_3\})}$$

Now $n(m_1, e_1), n(\{m_1, m_2\}, e_1)$, and $n(\{m_1, m_3\}, e_1)$ are again obtained from Table 2.10 as 11, 6 and 6 respectively. Similarly, $n(m_1), n(\{m_1, m_2\})$, and $n(\{m_1, m_3\})$ are obtained from the bottom row which indicates the total number of observations from each mode. All three of these values are equal to 14.

$$p(e_1|m_1,\Theta) = \frac{11 + \theta\{\frac{m_1}{m_1,m_2}\}6 + \theta\{\frac{m_1}{m_1,m_3}\}6}{14 + \theta\{\frac{m_1}{m_1,m_2}\}14 + \theta\{\frac{m_1}{m_1,m_3}\}14}$$

In more general terms, the expression for the likelihood can be given as

$$\begin{aligned} p(E|M,\Theta) &= \frac{\sum\limits_{\boldsymbol{m}_k \supseteq M} \theta\{\frac{M}{\boldsymbol{m}_k}\}n(\boldsymbol{m}_k, E)}{\sum\limits_{\boldsymbol{m}_k \supseteq M} \theta\{\frac{M}{\boldsymbol{m}_k}\}n(\boldsymbol{m}_k)} \\ &= \frac{\sum\limits_{\boldsymbol{m}_k \supseteq M} \theta\{\frac{M}{\boldsymbol{m}_k}\}n(\boldsymbol{m}_k)S(E|\boldsymbol{m}_k)}{\sum\limits_{\boldsymbol{m}_k \supseteq M} \theta\{\frac{M}{\boldsymbol{m}_k}\}n(\boldsymbol{m}_k)} \end{aligned} \tag{2.22}$$

where the support function $S(E|\boldsymbol{m}_k)$ can be obtained for discrete data as

$$S(E|\boldsymbol{m}_k) = \frac{n(\boldsymbol{m}_k, E)}{n(\boldsymbol{m}_k)}$$

2.4.3 Dempster-Shafer Theory

Dempster-Shafer theory was first proposed as a generalization to Bayesian methods in a manner that can account for uncertainty about the hypotheses (or for our purposes, the modes). Dempster-Shafer Theory has been developed as a framework for artificial intelligence with uncertain reasoning and many developments (particularly combination rules and methods of interpretation) have been made in this area. In this tutorial, the basics of Dempster-Shafer Theory from [50] will be covered.

Dempster-Shafer theory has two main proponents: the rule of conditioning and the rule of combination.

Dempster's Rule of Conditioning

Dempster's Rule of conditioning aims to find probability ranges as a method of describing the results. When the probability and likelihood were formulated earlier with Θ parameters, it was admitted that these Θ parameters were unknown. Because these parameters

are probabilities and must be contained on the interval of $[0, 1]$, it is possible to find the probability bounds by maximizing and minimizing over Θ. In [50], Dempster and Shaver never made use of parameters which would represent allocation of ambiguous mode data. Instead they were more concerned about probability ranges for the final result.

Dempster and Shafer first define the *support function* or *Basic Belief Assignment* which was briefly mentioned before when parametrizing probabilities using Θ. The support function $S(X)$ is defined in the same manner as probability but can be directly applied to ambiguous modes as well as unambiguous ones.

$$S(\boldsymbol{M}) = \frac{n(\boldsymbol{M})}{\sum_k n(\boldsymbol{m}_k)}$$
$$S(\boldsymbol{M}|E) = \frac{n(\boldsymbol{M}, E)}{n(E)}$$
$$S(E|\boldsymbol{M}) = \frac{n(\boldsymbol{M}, E)}{n(\boldsymbol{M})}$$

From the support function, we can calculate the lower bound and upper bound probabilities. In Demspter-Shafer theory, they are known as belief and plausibility.

1. $Bel(X)$ is the *Belief* or *lower bound probability* of X
2. $Pl(X)$ is the *Plausibility* or *upper bound probability* of X

The belief and plausibility can be obtained using *Demspter's Rule of Conditioning*

$$Bel(\boldsymbol{M}) = \sum_{\boldsymbol{m}_k \subseteq \boldsymbol{M}} S(\boldsymbol{m}_k) \tag{2.23}$$
$$Pl(\boldsymbol{M}) = \sum_{\boldsymbol{m}_k \cap \boldsymbol{M} \neq \emptyset} S(\boldsymbol{m}_k) \tag{2.24}$$

This rule of conditioning is consistent with our direct probability scenario. If we took the expression in Eqn (2.21) and minimized/maximized it over Θ, the result would be

$$Bel(M|E, \Theta) = \min_{\Theta} Bel(M|E, \Theta) = \sum_{\boldsymbol{m}_k = M} S(\boldsymbol{m}_k|E) \tag{2.25}$$
$$Pl(M|E, \Theta) = \max_{\Theta} Bel(M|E, \Theta) = \sum_{\boldsymbol{m}_k \supseteq M} S(\boldsymbol{m}_k|E) \tag{2.26}$$

The summation limits are slightly changed, but only because M is unambiguous. As one can see that

- If M is unambiguous $\boldsymbol{m}_k \subseteq M$ is only true when $\boldsymbol{m}_k = M$.
- If M is unambiguous, $\boldsymbol{m}_k \cap M \neq \emptyset$ is only true when $\boldsymbol{m}_k \supseteq M$.

The reason why Eqn (2.25) minimizes $P(M|E, \Theta)$ is because it is linear with respect to Θ and all of the coefficients on Θ are non-negative support functions $S(\boldsymbol{m}_k)$. Thus, by

minimizing the non-forced values of Θ, $P(M|E,\Theta)$ is also minimized. The only time the condition $\theta \equiv 1$ is forced is when θ is expressed as $\theta\{\frac{M}{M}\}$; all other terms can be zero if need be, making $\boldsymbol{m}_k = M$ the only term included in the belief summation condition. By the same reasoning, $P(M|E,\Theta)$ is maximized when Θ is maximized. Thus, we include in our summation every case where Θ is not forced to be zero. As long as $M \subseteq \boldsymbol{k}$ then $\theta\{\frac{M}{\boldsymbol{m}_k}\}$ is not forced to be zero, making $\theta\{\frac{M}{\boldsymbol{m}_k}\}$ the summation term for plausibility.

As an example, consider our previous case where we found that

$$\begin{aligned} p(m_1|e_1,\Theta) &= S(m_1|e_1)\theta\{\tfrac{m_1}{m_1}\} + S(\{m_1,m_2\}|e_1)\theta\{\tfrac{m_1}{m_1,m_2}\} + S(\{m_1,m_3\}|e_1)\theta\{\tfrac{m_1}{m_1,m_3}\} \\ &= \left[\tfrac{11}{29}\theta\{\tfrac{m_1}{m_1}\} + \tfrac{6}{29}\ \theta\{\tfrac{m_1}{m_1,m_2}\} + \tfrac{6}{29}\ \theta\{\tfrac{m_1}{m_1,m_3}\}\right] \end{aligned}$$

We can see that the value $\theta\{\frac{m_1}{m_1}\} = 1$ is forced by logic, but the other values $\theta\{\frac{m_1}{m_1,m_2}\}$ and $\theta\{\frac{m_1}{m_1,m_3}\}$ can be anywhere between 0 and 1. Because the coefficients on these two values are positive (they both equal $\frac{6}{29}$),

- $p(m_1|e_1,\Theta)$ is minimized when $\theta\{\frac{m_1}{m_1,m_2}\}$ and $\theta\{\frac{m_1}{m_1,m_3}\}$ are set to 0
- $p(m_1|e_1,\Theta)$ is maximized when $\theta\{\frac{m_1}{m_1,m_2}\}$ and $\theta\{\frac{m_1}{m_1,m_3}\}$ are set to 1

It should be noted however, that while Dempster's Rule of Conditioning functions for direct probabilities, it does not function for likelihoods. An adaptation of Dempster-Shafer theory that can be applied to likelihoods is formulated in later chapters (Chapters 5 and 9) pertaining to Generalized Dempster-Shafer Theory.

Dempster's Rule of Combination

In addition to the rule of conditioning, Dempster-Shafer theory also has a method to combine information from multiple *independent* sources, much like Bayesian methods. Dempster's Rule of Combination can be written as

$$\begin{aligned} S_{12}(\boldsymbol{m}_k) &= \frac{1}{1-K} \sum_{\boldsymbol{m}_k = \boldsymbol{m}_i \cap \boldsymbol{m}_j} S_1(\boldsymbol{m}_i) S_2(\boldsymbol{m}_j) \qquad \boldsymbol{m}_k \neq \emptyset \\ K &= \sum_{\emptyset = \boldsymbol{m}_i \cap \boldsymbol{m}_j} S(\boldsymbol{m}_i) S(\boldsymbol{m}_j) \end{aligned} \tag{2.27}$$

where $1-K$ is a normalization constant to ensure that $S(\boldsymbol{m}_k)$ sums to 1. Here, S_1 and S_2 must be support functions that come from independent sources and S_{12} is a support function that combines the two independent sources. If, for example, the bias monitor and stiction monitors could be considered independent, they could each be used to independently construct their own $S(\boldsymbol{m}_k|e)$. The two support functions could then be combined according to Eqn (2.27).

As a combination example, let us consider our system in Figure 2.3 with independent monitors. Let us say that $S(\boldsymbol{m}_k|e)$ is given as

$$\begin{aligned}
S_1(m_1) &= 4/16 & S_2(m_1) &= 3/16 \\
S_1(m_2) &= 2/16 & S_2(m_2) &= 2/16 \\
S_1(m_3) &= 1/16 & S_2(m_3) &= 2/16 \\
S_1(m_4) &= 1/16 & S_2(m_4) &= 1/16 \\
S_1(m_1, m_2) &= 3/16 & S_2(m_1, m_2) &= 2/16 \\
S_1(m_1, m_3) &= 2/16 & S_2(m_1, m_3) &= 3/16 \\
S_1(m_2, m_4) &= 2/16 & S_2(m_2, m_4) &= 2/16 \\
S_1(m_3, m_4) &= 1/16 & S_2(m_3, m_4) &= 1/16
\end{aligned}$$

If we wanted to combine these two BBAs to form S_{12} and assess $S_{12}(m_1)$, we would have

$$\begin{aligned}
S_{12}(m_1) &= \frac{1}{1-K} \sum_{\boldsymbol{m}_1 = \boldsymbol{m}_i \cap \boldsymbol{m}_j} S_1(\boldsymbol{m}_i) S_2(\boldsymbol{m}_j) \\
&= S_1(m_1)S_2(m_1) + S_1(m_1)S_2(m_1, m_2) + S_1(m_1, m_2)S_2(m_1) \\
&\quad + S_1(m_1)S_2(m_1, m_3) + S_1(m_1, m_3)S_2(m_1) \\
&\quad + S_1(m_1, m_2)S_2(m_1, m_3) + S_1(m_1, m_3)S_2(m_1, m_2) \\
&= \frac{1}{1-K} \left[\frac{4}{16} \cdot \frac{3}{16} + \frac{4}{16} \cdot \frac{2}{16} + \ldots + \frac{3}{16} \cdot \frac{3}{16} + \frac{2}{16} \cdot \frac{2}{16} \right]
\end{aligned}$$

As one can see, each multiplied pair contains sets that intersect to yield m_1.

2.5 Kernel Density Estimation

Up to this point, we have considered evidence E to be discrete. In most cases, the outputs of monitors are actually continuous, but discretization is performed in order to create individual alarms for each monitor. However, if we used smaller and smaller discretization regions to describe the likelihood $p(E|M)$,

$$\lim_{n(E) \to \infty} p(M|E) = \frac{f(E|M)p(M)}{\sum_M f(E|M)p(M)}$$

where $f(E|M)$ is the probability density function of the likelihood. If the type of distribution is known, (such as Gaussian), one could fit the data to this distribution using a parametric approach (such as maximum likelihood estimation). However, in most cases, monitor results do not follow a well-defined distribution, thus non-parametric methods need to be used. Namely, discretization or kernel density estimation [61].

2.5.1 From histograms to kernel density estimates

When trying to estimate a distribution from a data set without any knowledge of the distribution, most practitioners would turn to the histogram to perform this task. A histogram sections the data's domain into bins, and counts how many data points lie within each bin as shown in Figure 2.10.

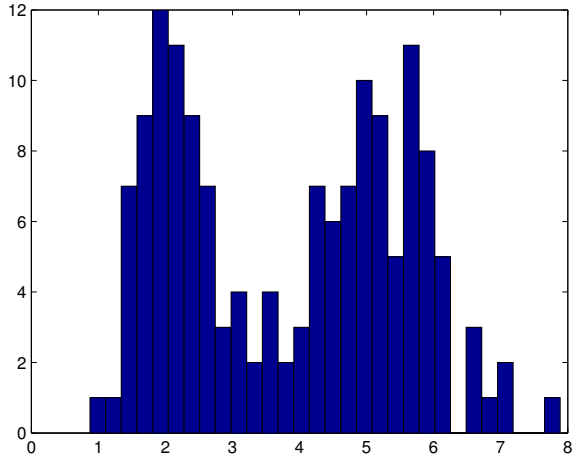


Figure 2.10: Histogram of distribution

The histogram is useful for getting a rough idea of what the distribution should look like. However, this sections data into bins and the cutoffs are fixed. A slight adaptation to the histogram is to center the counts around each data point. When evaluating the probability at a new point x, this centred histogram approach simply counts how many data points lie within a bin centred around x. Because the bin positions are more flexible, we can observe in Figure 2.11 that the distribution estimate is much smoother.

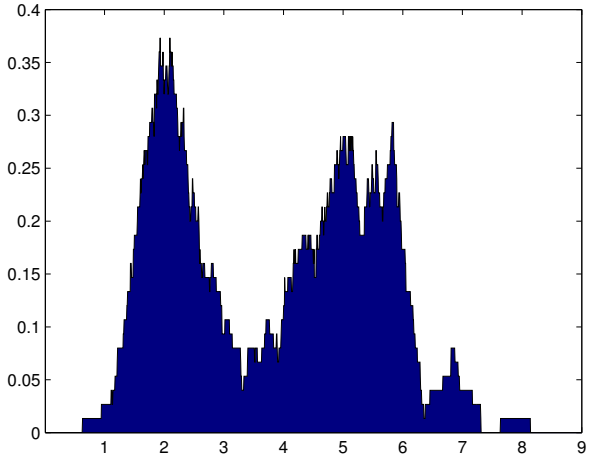


Figure 2.11: Centered histogram of distribution

The function represented by the centred histogram essentially places a block with area $1/n$ around each data point d_i (where n is the number of data points). When evaluating

the probability, we use the criterion

$$\hat{f}(x) = \frac{1}{n} \sum_{\substack{d_i \leq x+h \\ d_i \geq x-h}} \frac{1}{h}$$

where h is a boundary width set by the user. This is a basic form of a kernel density estimate. The general form of a kernel density estimate is

$$\hat{f}(x) = \frac{1}{n} \sum_i \frac{1}{h} K(x, d_i)$$

where in our example, the kernel function K is actually a block

$$K(x, d_i, h) \begin{cases} 0 & \{d_i < x - h\} \\ 1/h & \{x - h \leq d_i \leq x + h\} \\ 0 & \{d_i > x + h\} \end{cases}$$

Finally, after using this form, one realizes that the kernel function does not need to take the shape of a block, but can take on any desired shape, as long as it integrates to unity, for example, a standard Gaussian shape

$$K(x, d_i, h) = \frac{1}{h\sqrt{2\pi}} \exp\left[\frac{1}{2}\left(\frac{x - d_i}{h}\right)^2\right]$$

so that

$$\hat{f}(x) = \frac{1}{n} \sum_i \frac{1}{h\sqrt{2\pi}} \exp\left[\frac{1}{2}\left(\frac{x - d_i}{h}\right)^2\right] \tag{2.28}$$

If this kernel were applied to our data, the result would be a function resembling the one shown in Figure 2.12, which as one might observe, is much smoother.

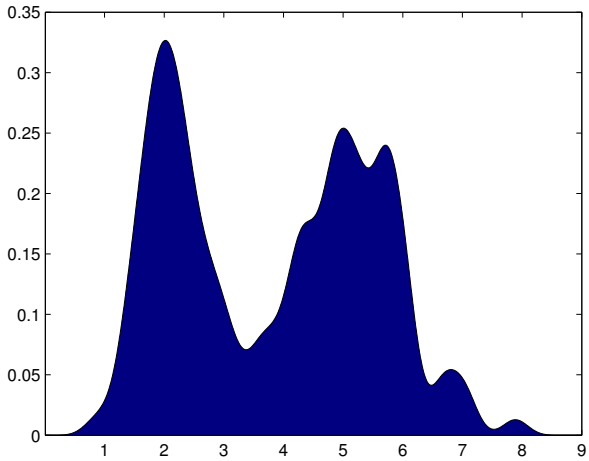


Figure 2.12: Gaussian kernel density estimate

Kernel density estimation thus far has been presented for univariate applications, but multivariate kernel density estimation is also possible. For example, a multivariate kernel density estimate could use the p-dimensional standard normal distribution to yield

$$\hat{f}(x) = \frac{1}{n}\sum_{i}\frac{1}{\sqrt{(2\pi)^p|H|}}\exp\left[\frac{1}{2}(x-d_i)^T H^{-1}(x-d_i)\right] \tag{2.29}$$

2.5.2 Bandwidth selection

Kernel density estimation itself is a simple procedure of summing up kernel functions centred around each data point, a procedure that has gone unchanged since its introduction in [61]. However, the main difficulty lies within the selection of the *bandwidth*, an area where research is ongoing. The bandwidth is synonymous to the bin-width in a histogram. Larger bandwidths will result in smoother kernel density estimates, while smaller bandwidths will result in rougher, more jagged estimates. Intuitively, a simple method of bandwidth selection is to start with a small bandwidth that yields a noisy, rough distribution and to use larger bandwidths until the distribution becomes smooth. This is a fairly labour-intensive approach, and hard to implement in dimensions larger than 2 (due to difficulties in visualization).

The goal of selecting a bandwidth is to minimize the error between the kernel density estimate and the true distribution. If the underlying distribution is known, an optimal kernel can be selected. The fundamentals in bandwidth selection have been presented quite nicely in [62], and the performance of adaptive bandwidth smoothing has been analyzed in-depth in [63]. While the optimal bandwidth can depend on the underlying distribution which we wish to estimate and the kernel type, it depends more strongly on the spread of the data points and the number of data points. Thus, fairly good results can be obtained by simply selecting a distribution (such as the Gaussian distribution) and using its optimal bandwidth estimate for other distributions. Because of this, the optimal Gaussian bandwidth estimator is commonly used

$$H_N = \left(\frac{4}{n(p+2)}\right)^{\frac{2}{p+4}} S \tag{2.30}$$

where S is the sample covariance estimate.

2.5.3 Kernel density estimation tutorial

Let us consider a set of generated data:

$$\mathcal{D} = \begin{bmatrix} 1.90 \\ 0.12 \\ 1.05 \\ -0.23 \\ -0.16 \\ 0.69 \\ 0.56 \\ -1.12 \\ -1.53 \\ -1.09 \end{bmatrix} \qquad \text{cov}(\mathcal{D}) = 1.1594$$

which can be visualized in Figure 2.13 where the data points lie along the x axis.

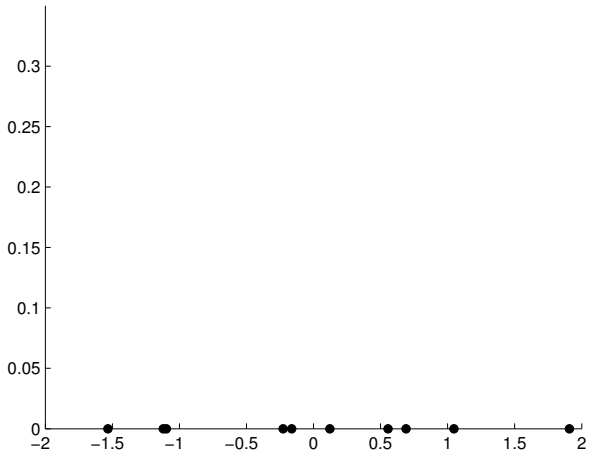


Figure 2.13: Data for kernel density estimation

Using the optimal Gaussian bandwidth selector in Eqn (2.30), we obtain a bandwidth of

$$\begin{aligned} H &= \left(\frac{4}{10(1+2)}\right)^{\frac{2}{1+4}} 1.1594 \\ &= \left(\frac{4}{10(1+2)}\right)^{\frac{2}{1+4}} 1.1594 \\ &= \left(\frac{4}{30}\right)^{\frac{2}{5}} 1.1594 \\ &= 0.5179 \end{aligned}$$

The individual kernels are shown around each data point in Figure 2.14

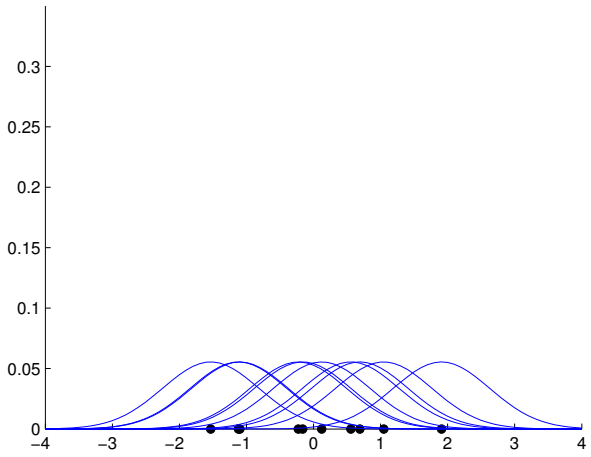


Figure 2.14: Data points with kernels

The bandwidth parameter can then be used in the kernel density estimate

$$\hat{f}(x) = \frac{1}{n}\sum_i \frac{1}{\sqrt{(2\pi)^p 0.5179}} \exp\left[\frac{1}{2}\frac{(x-d_i)^2}{0.5179}\right]$$

where d_i is each element of $\mathcal{D}$. This estimate is visualized in Figure (2.15)

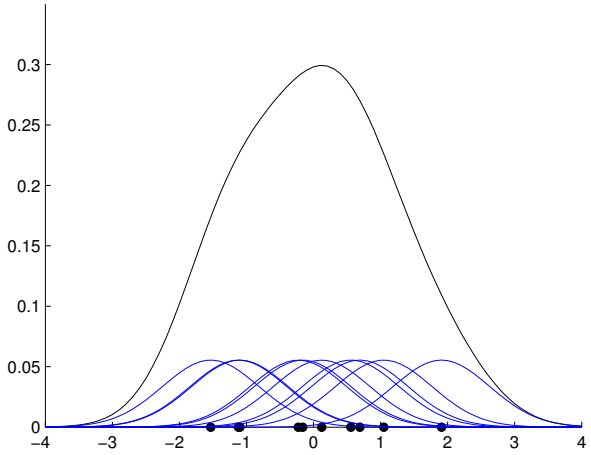


Figure 2.15: Kernel density estimate from data

2.6 Bootstrapping

Bootstrapping is a *resampling* method introduced by Efron [64] as a modification on jackknifing (another type of resampling technique). Resampling technques aim to construct

new sets of sampled data either by sampling subsets (as in jackknifing) or by sampling with replacement (as in Bootstrapping). Bootstrapping is particularly useful if one wishes to obtain a distribution about an estimated parameter. When bootstrapping, it is important that the data is independent and identically distributed (IID).

2.6.1 Bootstrapping tutorial

Let us consider a case where we have ten data points and we wish to estimate the mean.

$$\mathcal{D} = \begin{bmatrix} 6.68 \\ 3.83 \\ 4.14 \\ 5.51 \\ 5.33 \\ 4.56 \\ 4.05 \\ 4.77 \\ 4.16 \\ 5.06 \end{bmatrix} \qquad \hat{\mu} = 4.8090$$

However, we would also like to be able to determine a distribution around this estimate $\hat{\mu}$. Boostrapping enables us to perform this task. Firstly, one can create a new data set by drawing from the data at random with replacement. Equivalently, one can generate 10 numbers between 1 and 10

$$\text{IND} = [5, 2, 3, 3, 7, 8, 1, 4, 7, 9]$$

and then use these indices to define the new sample. For example, the first element of IND is 5. Thus the first element of the new data set is $\mathcal{D}(5) = 5.33$. The second element of IND is 2, thus the second element of the new data set is $\mathcal{D}(2) = 3.83$. This is continued until an entirely new data set of 10 entries is constructed

$$\mathcal{D}_1 = \begin{bmatrix} 5.33 \\ 3.83 \\ 4.14 \\ 4.14 \\ 4.05 \\ 4.77 \\ 6.68 \\ 5.51 \\ 4.05 \\ 4.16 \end{bmatrix} \qquad \hat{\mu}_1 = 4.6660$$

$\mathcal{D}_1$ is a resampling of $\mathcal{D}$ with replacement. Note that resampling yielded a slightly different value of $\hat{\mu}$. The resampling of $\mathcal{D}$ is then repeated until we have enough values of $\hat{\mu}$ to assess its distribution. For example, after resampling 1000 times, we have a distribution of $\hat{\mu}$ as shown in Figure 2.16.

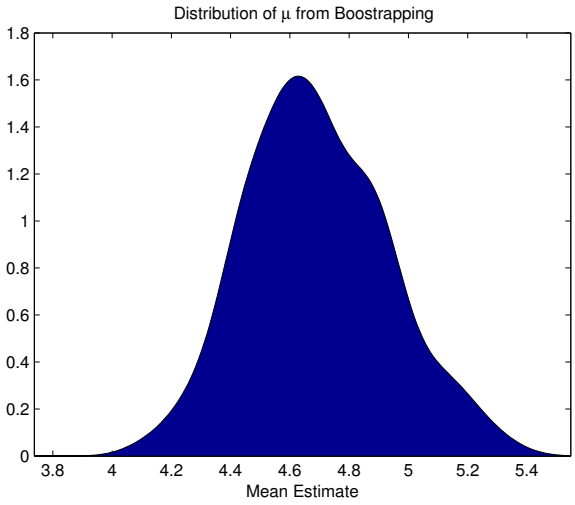


Figure 2.16: Distribution of $\hat{\mu}$ estimate

2.6.2 Smoothed bootstrapping tutorial

Resampling straight from the historical data can sometimes have strange and undesirable properties, such as there being a limited number of possible bootstrap sets. For example, from our 10 data sets, we could only have 10^{10} different data sets, but since ordering does not matter (because of $\hat{\mu}$ involves a sum), there would only be about

$$\frac{(n+r-1)!}{r!(n-1)!} = \frac{(10+10-1)!}{10!(10-1)!} = \frac{21!}{10!9!} = 38798760$$

unique values of $\hat{\mu}$ that could be sampled from. This may seem like a large number, but it would not take very many samples before there are a number of exact repetitions of bootstrapped values of $\hat{\mu}$.

In this example, ordinary bootstrapping would be sampling from a distribution shown in Figure 2.17. As one might see, the resampling can only take on a finite number of values, and the probability is zero for all other legitimate values that the data could have taken.

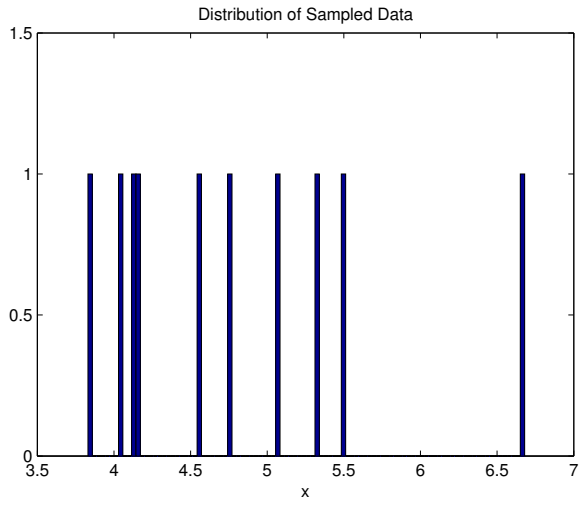


Figure 2.17: Sampling distribution for bootstrapping

In order to increase the number of values that $\hat{\mu}$ could take, adding Gaussian noise to each resampled piece of data would be an intuitive solution. This is called the *smoothed bootstrap*. However, adding Gaussian noise to the resampled data begs the question, what variance should this Gaussian noise have? Fortunately, this can be answered using kernel density estimation.

As mentioned in Section 2.5, kernel density estimation (or kernel smoothing) is a method to estimate a smoothed distribution from sampled data. By performing kernel smoothing on the data shown in Figure 2.17. This distribution comes from summing Gaussian distributions centred at each data point

$$\hat{f}(x) = \frac{1}{n} \sum_i \frac{1}{\sqrt{(2\pi)^p |H|}} \exp\left[\frac{1}{2}(x - d_i)^T H^{-1} (x - d_i)\right] \tag{2.31}$$

where H is the bandwidth matrix, which can be calculated as

$$H = \left(\frac{4}{n(p+2)}\right)^{\frac{2}{p+4}} S \tag{2.32}$$

Here, p is the dimension of the data, n is the number of data points and S is the sample covariance from the data. This results in the smoothed distribution shown in Figure 2.18. Note that this distribution (from Eqn (2.31)) can be sampled by randomly selecting one of the historical data points, and adding Gaussian noise with covariance H. This is exactly the same procedure as the smoothed bootstrap.

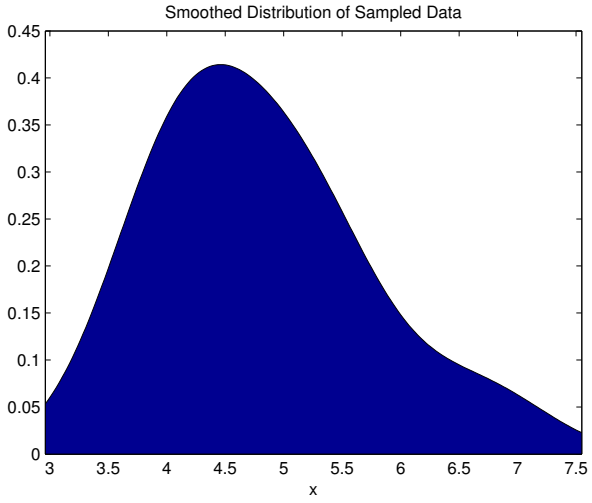


Figure 2.18: Smoothed sampling distribution for bootstrapping

Hence, by using the optimal bandwidth in Eqn (2.32), we can obtain appropriate covariance for the Gaussian noise to be added from our bootstrapped samples.

As an example, let us say we observed the same 10 data points for which we bootstrapped previously, and again we generated the selection indices

$$\text{IND} = [5, 2, 3, 3, 7, 8, 1, 4, 7, 9]$$

The covariance of the data was found to be $\text{var}(\mathcal{D}) = 0.75481$, resulting in a bandwidth of

$$\begin{aligned} H &= \left(\frac{4}{10(1+2)}\right)^{\frac{2}{1+4}} S \\ &= \left(\frac{4}{30}\right)^{\frac{2}{5}} 0.75481 \\ &= 0.33714 \end{aligned}$$

We can then add Gaussian noise to the bootstrapped sample

$$\mathcal{D}_1 = \begin{bmatrix} 5.33 \\ 3.83 \\ 4.14 \\ 4.14 \\ 4.05 \\ 4.77 \\ 6.68 \\ 5.51 \\ 4.05 \\ 4.16 \end{bmatrix} + \sqrt{0.33714} \times \texttt{randn}(10, 1)$$

where $\texttt{randn}(10, 1)$ generates a 10×1 vector of Gaussian random variables. The mean of $\mathcal{D}_1$ can be calculated to generate a resampled estimate of $\hat{\mu}_1$, but this time $\hat{\mu}_1$ can essentially

take on an infinite number of different values. The resulting distribution of $\hat{\mu}$ is shown in Figure 2.19 from which confidence intervals of $\hat{\mu}$ can be obtained.

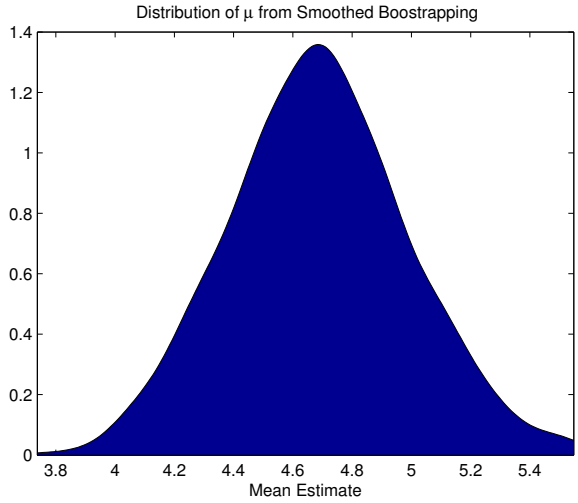


Figure 2.19: Distribution of $\hat{\mu}$ estimate

Chapter 3

Introduction to Testbed Systems

3.1 Simulated System

The system used for simulation is the well-known Tennessee Eastman Process [65] which has been the testbed for a host of process control and fault-diagnosis techniques [66] [67] [68] [48] (The schematic is presented in Figure 3.1). In this system, the gaseous reactants A, D, and E are fed directly into the reactor along with intert gas B (C comes in through the recycle stream). In this reactor, G and H (liquids at standard conditions) are formed according to the following reactions

$$A(g) + C(g) + D(g) \rightarrow G(l)$$
$$A(g) + C(g) + E(g) \rightarrow H(l)$$

which are irreversible, exothermic, and approximately first-order with respect to reactant concentrations. Product from the reactor is condensed and sent to a separator to remove the reactants (which are much more volatile than the products). Liquid separator product is then stripped using reactant C as the stripping agent. Liquid product from the stripper is seen as the final product. Meanwhile, gaseous product form the stripper and separator are recycled to the reactor, with some of the separator product being purged in order to prevent build-up of inert gas B.

The code used for simulation allows for 15 known pre-programmed process faults and uses the decentralized control strategy outlined by [69] (code is available in the Tennessee Eastman Challenge Archive [70]). For our applications, we consider a normal operating mode, and seven other modes where each fault happens one at a time (modes with multiple faults are not considered). A list of these modes is presented in Table 3.1.

3.1.1 Monitor design

In order to assess improvements shown by the Bayesian approach, monitors are chosen arbitrarily, some of which have high false-alarm/misdetection rates. By selecting monitors with mediocre performance, the merits of our proposed methods can be more clearly seen.

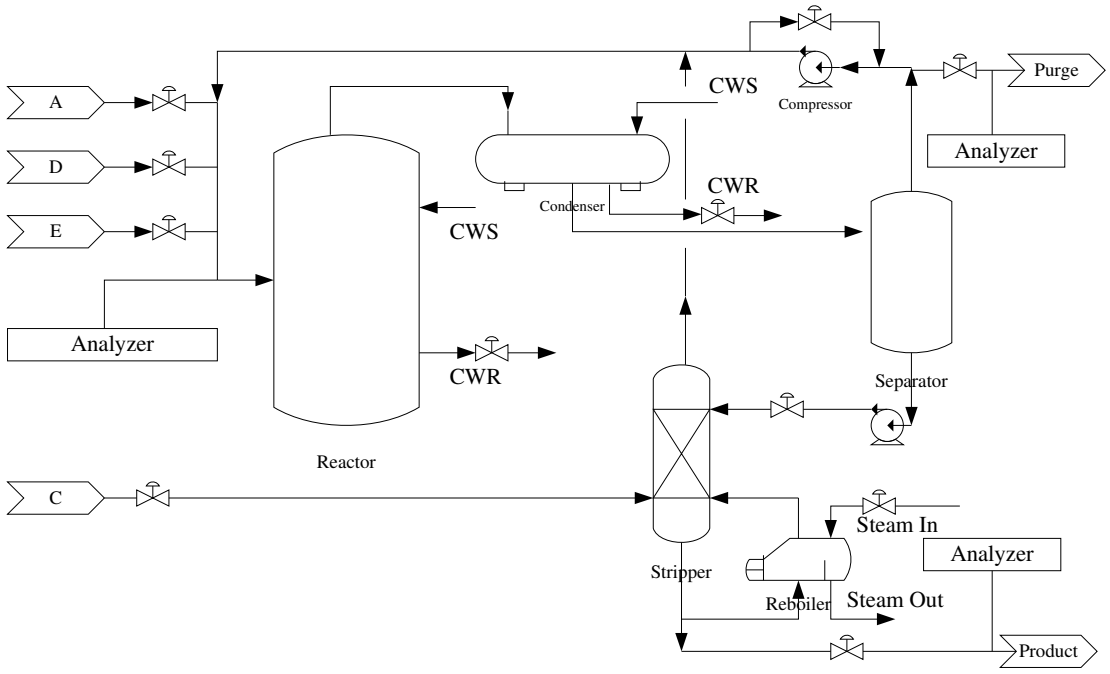


Figure 3.1: Tennessee Eastman process

Table 3.1: List of simulated modes

Variable number	Process variable	Type
NF	N/A	N/A
IDV 1	A/C feed ratio B composition constant (stream 4)	Step
IDV 2	B composition, A/C ratio constant (stream 4)	Step
IDV 7	C header pressure loss, reduced availability	Step
IDV 8	A, B, C feed composition (stream 4)	Variation
IDV 9	D feed temperature (stream 2)	Variation
IDV 12	Reactor cooling water inlet temperature	Variation
IDV 14	Reactor cooling water valve	Sticking

Control performance monitor

Six univariate control performance monitors are commissioned to monitor the control performance of the six key PVs. The FCOR algorithm [34] is employed to compute control performance indices based on univariate CVs.

Valve stiction monitor

According to Downs and Vogel [65], the reactor cooling water valve and the condenser cooling water valve both have the potential to develop stiction. Two valve stiction monitors

are commissioned to monitor these problems.

For illustrative purposes, we consider the following simplified scenario: if a control loop has oscillation, then the oscillation is caused either by valve stiction or by an external oscillatory disturbance. The latter has sinusoid form while the former does not.

If the CV and the MV of a control loop oscillate sinusoidally, an ellipse will be obtained when plotting CV versus MV. It has been observed that an ellipse will be distorted if the oscillation is caused by valve stiction. The method adopted here is based on the evaluation of how well the shape of the CV versus MV plot can be fitted by an ellipse. An empirical threshold of distance between each data point and the ellipse is used to determine the goodness-of-fit, and thereafter the existence of valve stiction.

Process model validation monitor

In addition to the control performance monitors, three additional model validation monitors are commissioned to monitor model changes for the reactor, separator and stripper levels.

The local approach based on the output error (OE) method [71] is employed to validate the nominal process model. This method applies to MISO systems (note that any MIMO system can be separated into several MISO subsystems). Models of each MISO part can be monitored with the local approach.

3.2 Bench Scale System

The hybrid tank system is a bench-scale process with a schematic given in Figure 3.2. The system consists of three tanks that can be interconnected through a system of valves. The two outer tanks have water inlets that can be used to control the water level, while all three tanks have outlets that empty into the main water basin. In this thesis, only the lower valves (valves 1 and 2) are used to interconnect the tanks. Furthermore, the outlet valve for the middle tank (valve 8) is closed, while the other outlets remain open. Valve 1 and Valve 2 are opened and closed in order to create two states for each of these components. In addition, bias is added to flow meters 1 and 2 in order to create states for two more components. When coupled with the open and closed states for the two valves, the system has $2^4 = 16$ possible operating modes.

The system is equipped with seven different measurements. Flow meters 1 and 2 were already mentioned; these instruments measure water flow rates into tanks 1 and 2 respectively. Control signals to the pumps are also measured (for pumps 1 and 2), as well as the three tank levels (level transmitters 1, 2 and 3).

Detection of different operating modes is done using model validation. A gray box model

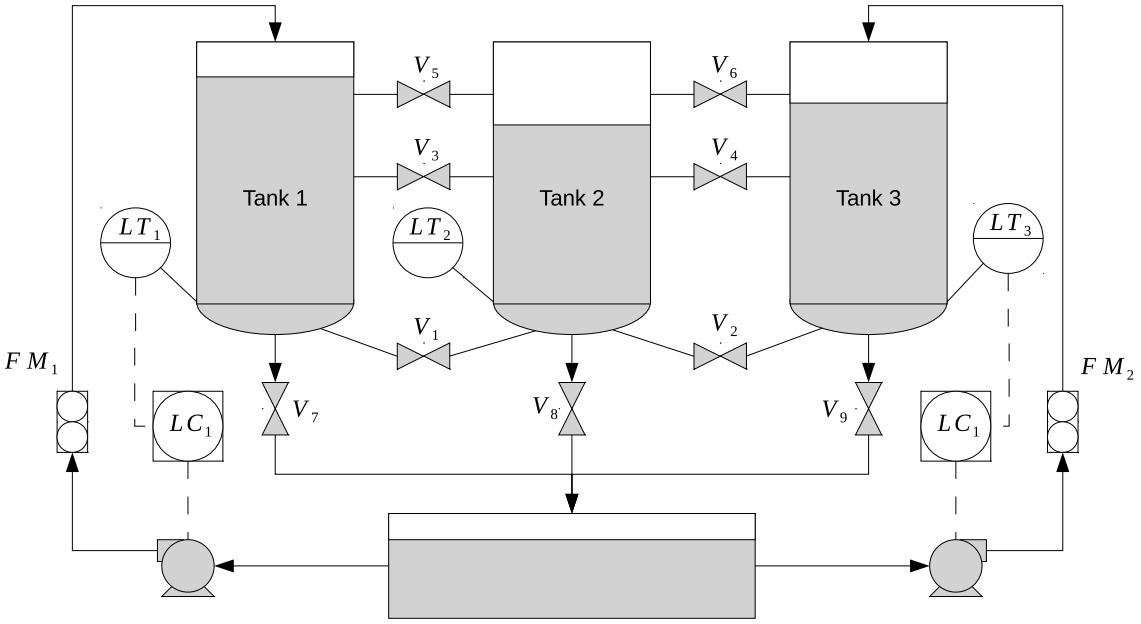


Figure 3.2: Hybrid tank system

is obtained for each tank, given as

$$\frac{d\ L_1}{d\ t} = A_c^{-1}\left(F_1 - C_1 L_1^{1/2}\right) \tag{3.1}$$

$$\frac{d\ L_2}{d\ t} = -A_c^{-1}\left(C_2 L_2^{1/2}\right) \tag{3.2}$$

$$\frac{d\ L_3}{d\ t} = A_c^{-1}\left(F_2 - C_3 L_3^{1/2}\right) \tag{3.3}$$

where A_c is the cross-sectional area of the tanks, F_1, F_2 are flow rates into tanks 1 and 3 respectively, C_1, C_2, C_3 are flow coefficients for each tank outlet, and L_2, L_2, L_3 are tank levels for each of the tanks.

Once this model is estimated, it is perturbed in order to include bias terms B_1, B_2 for the flow rate measurements of water going into tanks 1 and 3, and leak terms C_{L_1} and C_{L_2} which are flow constants for valves 1 and 2 when they are open.

$$\frac{d\ L_1}{d\ t} = A_c^{-1}\left[B_1 F_1 - C_1 L_1^{1/2} + C_{L_1} F_L(L_1, L_2)\right]$$

$$\frac{d\ L_2}{d\ t} = A_c^{-1}\left[-C_2 L_2^{1/2} + C_{L_1} F_L(L_2, L_1) + C_{L_2} F_L(L_2, L_3)\right]$$

$$\frac{d\ L_3}{d\ t} = A_c^{-1}\left[B_2 F_2 - C_3 L_3^{1/2} + C_{L_2} F_L(L_3, L_2)\right]$$

where $F_L(L_1, L_2)$ is a function given as

$$F_L(L_1, L_2) = (L_2 - L_1)|L_2 - L_1|^{-1/2}$$

Using the unscented Kalman filter, the monitor estimates parameters $B_1, B_2, C_{L_1}, C_{L_2}$ as additional hidden states.

In addition, the model between pump signal and measured flow rate gives additional estimates for bias terms B_1, B_2. The estimate is given as

$$B_1 = \frac{\hat{F}_1(U_1)}{F_1}$$
$$B_2 = \frac{\hat{F}_2(U_2)}{F_2}$$

where U_1 and U_2 are pump control signals to pumps 1 and 2 respectively, and $\hat{F}_1(U_1), \hat{F}_2(U_2)$ are the respective flow predictions based on pump signals. When combined with level measurements, there is a total of nine monitoring inputs to this system.

3.3 Industrial Scale System

The industrial system considered for examples in this thesis is part of a solids handling facility in Canada's oil sands industry, similar to one used by Gonzalez and Huang [54]. As part of the preparation, mined oil sand is crushed, sized, and made into slurry though the breaker system. Data obtained from this facility does not have faults, but each of the subsystems (the size, conveyor and breaker) each operates under the modes "on" and "off"; this leads to eight possible modes overall. However, certain modes occur very infrequently due to safety systems installed; namely, the conveyor is turned off if the breaker is turned off, and the sizer is turned off when the conveyor is turned off. Nevertheless, as rare as these modes can be, they occur in history during transitional periods.

The system makes use of twenty unique measurements; the measurements along with a simple system schematic are given in Figure 3.3.

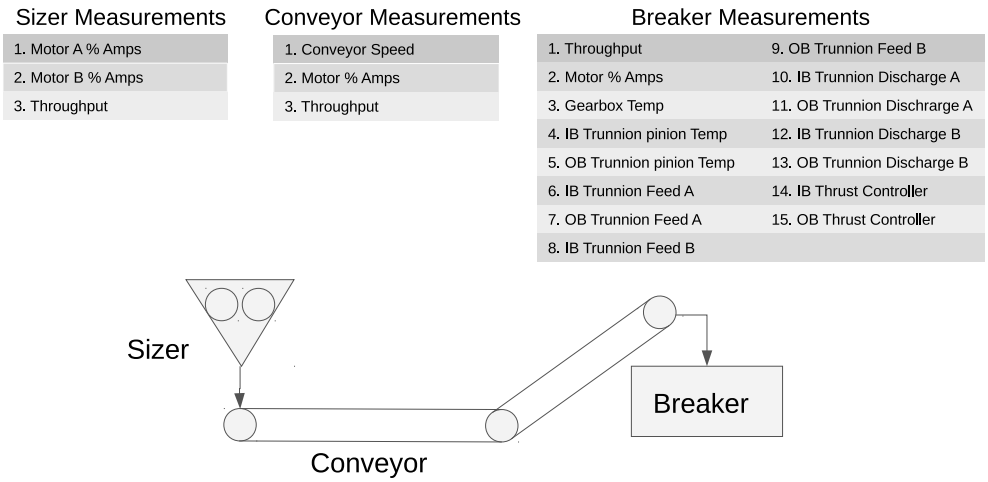


Figure 3.3: Solids handling system

Chapter 4

Accounting for Ambiguous Modes: A Bayesian Approach

4.1 Introduction

The first challenge to be addressed in this thesis is the problem of missing information about the mode (a challenge similar to *missing information about the evidence* which was addressed by Qi and Huang [1]).

As indicated in Chapter 2, a mode comprises of a set of states for individual components. For example, consider a system with two components c_1, c_2; let c_1 represent a sensor and c_2 represent a valve. A mode must have information for both components $[c_1, c_2]$. If, for this system, the sensor has three states, (positive bias, no bias, and negative bias), while the valve has three states, (no stiction, moderate stiction, and severe stiction) there is a total of nine possible modes.

When information about any of the system components is missing, the mode is said to be ambiguous. For example, let us say that in a certain segment of historical data, we are unsure of the state of the valve, but we know there is moderate stiction, the corresponding mode is $[\ c_1 = \times\ ,\ c_2 = 2\ ]$. The mode is ambiguous as it could be one of three specific modes $[\ c_1 = 1\ ,\ c_2 = 2\ ], [\ c_1 = 2\ ,\ c_2 = 2\ ], [\ c_1 = 3\ ,\ c_2 = 2\ ]$. The aim of this chapter is how to deal with ambiguous modes, such as this, when they appear in the data.

4.2 Parametrization of Likelihoods Given Ambiguous Modes

4.2.1 Interpretation of proportion parameters

When data in the history is taken from an ambiguous mode, a proportion of this data may belong to any of the specific modes within the ambiguous mode; for example, the specific mode $[\ c_1 = 1\ ,\ c_2 = 2\ ]$ within the ambiguous mode $[\ c_1 = \times\ ,\ c_2 = 2\ ]$. In order to deal with ambiguity, we consider a set of unknown parameters $\theta\{\frac{M}{\boldsymbol{m}}\}$ which indicate the proportion of data under the potentially ambiguous mode $\boldsymbol{m}$ belonging to any of the specific modes $M \subseteq \boldsymbol{m}$. For example, if we had 10 data points for mode $[\ c_1 = \times\ ,\ c_2 = 2\ ]$, and

we knew that three of them belonged to mode [$1 = \times$, $c_2 = 2$], then the corresponding θ parameter would be assigned a value of $\theta\{\frac{[1,2]}{[\times,2]}\} = 3/10$. Note that in this thesis, the boldface $\boldsymbol{M}$ can indicate any observed mode that has occurred in the data (including an ambiguous one) but M (or m, if we are talking about observations) can only represent a specific (unambiguous) mode.

When expressed as a probability, $\theta\{\frac{M}{\boldsymbol{m}}\}$ is equated to

$$\theta\{\tfrac{M}{\boldsymbol{m}}\} = p(M|\boldsymbol{m})$$

which is the probability that the specific mode M occurs given the potentially ambiguous mode $\boldsymbol{m}$.

As a practical example of defining θ, let us consider the same system with the valve and sensor each being able to take on three states. The resulting modes are as follows:

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \\ m_7 \\ m_8 \\ m_9 \end{bmatrix} = \begin{bmatrix} c_1 = 1 \ , \ c_2 = 1 \\ c_1 = 1 \ , \ c_2 = 2 \\ c_1 = 1 \ , \ c_2 = 3 \\ c_1 = 2 \ , \ c_2 = 1 \\ c_1 = 2 \ , \ c_2 = 2 \\ c_1 = 2 \ , \ c_2 = 3 \\ c_1 = 3 \ , \ c_2 = 1 \\ c_1 = 3 \ , \ c_2 = 2 \\ c_1 = 3 \ , \ c_2 = 3 \end{bmatrix}$$

If we consider data coming from the ambiguous mode [$c_1 = \times$, $c_2 = 2$], or equivalently, the set of modes $\{m_2, m_5, m_8\}$, a certain proportion $\theta\{\frac{m_2}{m_2,m_5,m_8}\}$ of that data actually belongs to mode m_2, another proportion $\theta\{\frac{m_5}{m_2,m_5,m_8}\}$ belongs to m_5, a final proportion of the data $\theta\{\frac{m_8}{m_2,m_5,m_8}\}$ belongs to m_8. Each member θ in Θ (which contains all θ values) is an unknown quantity unless the following conditions apply:

$$\theta\{\tfrac{m_i \not\subseteq \boldsymbol{m}_k}{\boldsymbol{m}_k}\} = 0 \qquad \text{e.g. } \theta\{\tfrac{m_2}{m_1,m_4,m_7}\} = 0 \tag{4.1}$$

$$\theta\{\tfrac{m_i = \boldsymbol{m}_k}{\boldsymbol{m}_k}\} = 1 \qquad \text{e.g. } \theta\{\tfrac{m_2}{m_2}\} = 1 \tag{4.2}$$

For the first exception, the mode m_2 is not possible given the ambiguous mode $\{m_1, m_4, m_7\}$, thus none of the data from this ambiguous mode can belong to mode m_2. For the second exception, given the mode m_2, all of the data in m_2 will belong to m_2, it cannot belong anywhere else; in a probabilistic sense, it would be equivalent to stating $p(M = m_2|m_2) = 1$ (or $p(m_2|m_2) = 1$ for shorthand). These values of Θ, being predetermined by logic are *logically forced*.

4.2.2 Parametrizing likelihoods

The principal data-driven component of Bayesian inference is the likelihood. When combined with prior probabilities, the posterior is calculated as

$$p(M|E) = \frac{p(E|M)p(M)}{p(E)} \tag{4.3}$$

$$p(E) = \sum_{M} p(E|M)p(M) \tag{4.4}$$

When ambiguous modes are present, obtaining likelihoods $p(E|M)$ can be quite challenging, as the conditioning variable M is unknown in ambiguous mode circumstances. In the discrete case, when no prior samples are taken, the likelihood is obtained as

$$p(E|M) = \frac{n(E, M)}{n(M)}$$

However, if we are calculating $p(E|m_1)$ and there is an ambiguous mode $\{m_1, m_2\}$ in the data, then we need to take into account the data in $\{m_1, m_2\}$ that belongs to mode m_1. Thus, the likelihood is calculated as

$$p(E|m_1) = \frac{n(E, m_1) + \theta\{\frac{m_1}{m_1, m_2}\}n(E, \{m_1, m_2\})}{n(m_1) + \theta\{\frac{m_1}{m_1, m_2}\}n\{m_1, m_2\}}$$

Let us now revisit the term, *support* as given in [50], which is calculated in the same manner as probability, except that now ambiguous modes can be used as the conditioning variable

$$S(E|\boldsymbol{M}) = \frac{n(E, \boldsymbol{M})}{n(\boldsymbol{M})}$$

where, again, the boldface $\boldsymbol{M}$ indicates that the mode can be ambiguous. The previous likelihood expression can then be given as

$$p(E|m_1) = \frac{n(E, m_1) + \theta\{\frac{m_1}{m_1, m_2}\}S(E|\{m_1, m_2\})n\{m_1, m_2\}}{n(m_1) + \theta\{\frac{m_1}{m_1, m_2}\}n\{m_1, m_2\}}$$

In this chapter, it is assumed that $S(E|\boldsymbol{M})$ is derived from discrete evidence, but continuous evidence can be used in its stead as well; note that the utilization of continuous evidence is addressed in Chapter 6.

The expression for the likelihood $p(E|M, \Theta)$ accounted for only one ambiguous mode, but in general the likelihood can be expressed to take into account multiple ambiguous modes:

$$p(E|M, \Theta) = \frac{\sum\limits_{\boldsymbol{M} \supseteq M} \theta\{\frac{M}{\boldsymbol{M}}\}S(E|\boldsymbol{M})n(\boldsymbol{M})}{\sum\limits_{\boldsymbol{M} \supseteq M} \theta\{\frac{M}{\boldsymbol{M}}\}n(\boldsymbol{M})} \tag{4.5}$$

where the summation limit $\boldsymbol{M} \supseteq M$ cycles through all historical modes and includes every historical mode $\boldsymbol{m}_k$ that can support the mode in question M. Note that the term Θ includes the variables in Θ that can support the mode M; other variables in Θ do not play any role in calculating $p(E|M)$.

4.2.3 Informed estimates of likelihoods

As with Dempster-Shafer theory (mentioned before in Chapter 2), the presence of ambiguity in the historical data will result in probability ranges, as the likelihoods depend on variables in Θ. The lower-bound probability is called *Belief* while the upper-bound probability is called *Plausibility*

$$Bel(E|M) = \min_{\Theta} p(E|M, \Theta) \tag{4.6}$$

$$Pl(E|M) = \max_{\Theta} p(E|M, \Theta) \tag{4.7}$$

In addition to the belief and plausibility, there is also a probability between these two boundaries that represents the best estimate of the likelihood. This best estimate can be obtained by using prior probabilities. For example, let us consider an ambiguous mode $\{m_1, m_2\}$. If there is complete ignorance as to whether any of the data belongs to mode m_1 or mode m_2, then the data should be divided evenly according to the principle of insufficient reason. However, if there is prior knowledge that mode m_1 happens three times as frequently as mode m_2, $(p(m_1) = 3p(m_2))$, then it stands to reason that 75 % of the data in $\{m_1, m_2\}$ belongs to m_1 and 25 % belongs to m_2. This would equate to the following parameters:

$$\hat{\theta}\{\tfrac{m_1}{m_1,m_2}\} = p(m_1|\{m_1, m_2\}) = 0.75$$
$$\hat{\theta}\{\tfrac{m_2}{m_1,m_2}\} = p(m_2|\{m_1, m_2\}) = 0.25$$

These estimates of $\theta\{\frac{m_1}{m_1,m_2}\}$ and $\theta\{\frac{m_2}{m_1,m_2}\}$ are called *informed* estimates as they make use of prior information. In general, the informed estimate of a proportion parameter is given as

$$\hat{\theta}\{\tfrac{M}{\boldsymbol{m}}\} = \frac{p(M)}{\sum\limits_{M \subseteq \boldsymbol{m}} p(M)} \tag{4.8}$$

where the summation limit $M \subseteq \boldsymbol{m}$ cycles through all unambiguous modes M that are contained within the historical (and potentially ambiguous) mode $\boldsymbol{m}$. Note that logically, informed estimates in $\hat{\Theta}$ are also subject to the conditions in Eqn (4.1) and (4.2). The informed likelihood can serve as an educated guess as to what the likelihood value should be and is important for techniques mentioned later on in this chapter.

4.3 Fagin-Halpern Combination

While a parametrized expression now exists for the likelihood in Eqn (4.5), applying Bayes' rule in Eqn (4.3) is quite difficult, and successive combinations will yield more complex results. Fagin and Halpern [72] proposed a conditioning rule that can be used to combine

likelihood ranges with prior probabilities.

$$Bel(M|E) = \frac{Bel(M,e)}{Bel(M,E) + Pl(\bar{M},E)}$$
$$Pl(M|E) = \frac{Pl(M,e)}{Pl(M,E) + Bel(\bar{M},E)}$$

where $\bar{M}$ denotes all specific modes that are not M. When applied to Bayes' rule of combination, the Fagin Halpern combination rule can be given as

$$Bel(M|E) = \frac{Bel(E|M)Bel(M)}{Bel(E|M)Bel(M) + \sum_{\bar{M}} Pl(E|\bar{M})Pl(\bar{M})} \tag{4.9}$$

$$Pl(M|E) = \frac{Pl(E|M)Pl(M)}{Pl(E|M)Pl(M) + \sum_{\bar{M}} Bel(E|\bar{M})Bel(\bar{M})} \tag{4.10}$$

This rule gives the largest possible boundary, and is suitable for combining a single likelihood term with a prior probability. The difficulty is that when successive combinations are used (which can happen when applying this method dynamically), the Fagin Halpern rule yields boundaries that quickly grow to a point where the end result is uninformative.

For example, if we consider evidence E where certain elements E_1 are independent of the other elements E_2, ($E = [E_1, E_2], \quad E_1 \perp E_2$), then

$$p(E|M) = p(E_1|M)p(E_2|M)$$

When we consider ambiguous modes, $p(E|M,\Theta)$, the same applies

$$p(E|M,\Theta) = p(E_1|M,\Theta)p(E_2|M,\Theta)$$

If we use Fagin Halpern combination to combine the two results

$$Bel(E|M) = \frac{Bel(E_1|M)Bel(E_2|M)}{Bel(E_1|M)Bel(E_2|M) + \sum_{\bar{M}} Pl(E_1|\bar{M})Pl(E_1|\bar{M})}$$
$$Pl(E|M) = \frac{Pl(E_1|M)Pl(E_2|M)}{Pl(E_1|M)Pl(E_2|M) + \sum_{\bar{M}} Bel(E_1|\bar{M})Bel(E_2|\bar{M})}$$

the end result would yield probability boundaries larger than the original boundaries by maximizing and minimizing the original expression $p(E|M,\Theta)$. As a result, Fagin Halpern boundaries are too conservative for separating likelihoods, and are also too conservative for sequential combination as done in dynamic applications.

4.4 Second-Order Approximation

Up to this point, we have considered two solutions:

1. Directly applying the Bayesian solution in Eqn (4.3) to the parametrized likelihood in Eqn (4.5), but it was found that the expression grew successively more complicated with each combination step

2. Applying the Fagin-Halpern method in Eqn (4.9) and (4.10). This was found to be suitable for combining a single likelihood with a belief, but since probability boundaries grew with each successive combination, it was found to be unsuitable for separating evidence into independent groups and for dynamic application.

In order to retain some of the properties of the first (direct) solution while maintaining fixed simplicity like the second (Fagin-Halpern) solution, we propose a second-order approximation. We can first express Eqn (4.5) as a second-order function, and perform combination, ignoring higher-order terms. Thus the parametrized likelihood will remain to be a second-order expression with respect to Θ. While this is an approximate solution, due to the fact that the domain of Θ is restricted to values between 0 and 1, there is generally not enough room in the domain to deviate strongly from second-order behaviour. Hence, the second-order approximation is reasonable throughout the entire domain of $p(E|M,\Theta)$. Finally, the second-order apprixmaiton is exact at its reference point. In this case, we set the reference point for Θ to be the *informed* probability, which is our best guess at the value for Θ. Thus the second-order approximation inherently makes use of a best-guess probability that can be easily obtained from this method.

4.4.1 Consistency of Θ parameters

One important advantage of the second-order combination rule has over the Fagin-Halpern combination method is the ability to assume consistent Θ parameters. Let us consider a case where we would like to separate $p(E|M,\Theta)$ into two independent distributions so that

$$p(E|M,\Theta) = p(E_1|M,\Theta)p(E_2|M,\Theta) \tag{4.11}$$

For the Fagin-Halpern method of combination, it is assumed that Θ can take on different values for E_1 than E_2. In making this assumption, the probability boundaries are grown in such a way as to encompass all possible probability results from all values of Θ given that they are allowed to independently vary for E_1 and E_2. In reality, the values of Θ used for E_1 must be the same values used for E_2 if the independent combination in Eqn (4.11) is to hold true.

Whether directly combining independent likelihoods $p(E_1|M,\Theta)$ and $p(E_2|M,\Theta)$, or combining their approximation, because all values of Θ must be the same between the two sources of evidence, the terms in Θ can be collected. Thus, when applying the combination rule to second-order approximations, zeroth, first and second order terms of Θ are collected, and higher order terms are ignored.

When evaluating evidence at different time intervals $p(E^t|M^t,\Theta)$, the values of Θ do not change with time because same historical data with the same values of Θ are used to evaluate

the likelihood at each different time step t. Thus, not only is collecting Θ terms valid when combining likelihoods form independent evidence, it is also valid when applying the second-order rule in a dynamic fashion. Dynamic application of the second-order combination rule will be covered later on in this chapter.

4.4.2 Obtaining a second-order approximation

The second-order approximation of a function $f(x)$ is given by the Taylor Series

$$f(\boldsymbol{x}) \approx f(\hat{\boldsymbol{x}}) + \boldsymbol{J}(\boldsymbol{x} - \hat{\boldsymbol{x}}) + \frac{1}{2}(\boldsymbol{x} - \hat{\boldsymbol{x}})^T \boldsymbol{H}(\boldsymbol{x} - \hat{\boldsymbol{x}})$$

where $\hat{\boldsymbol{x}}$ is a reference point around which the approximation is taken (the approximation is exact at $\hat{\boldsymbol{x}}$ but becomes worse when x is further away from $\hat{\boldsymbol{x}}$), $\boldsymbol{J}$ is the Jacobian matrix (first-order derivatives evaluated at $\hat{\boldsymbol{x}}$)

$$\boldsymbol{J}_i = \left.\frac{\partial f(\boldsymbol{x})}{\partial x_i}\right|_{\hat{x}}$$

and $\boldsymbol{H}$ is the Hessian matrix (second-order derivatives also evaluated at $\hat{\boldsymbol{x}}$)

$$\boldsymbol{H}_{i,j} = \left.\frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \, \partial x_j}\right|_{\hat{x}}$$

When applied to our problem, the second order approximation of $p(E|M,\Theta)$ is calculated with respect to Θ. A convenient reference point for Θ is the informed estimate $\hat{\Theta}$. For the Jacobian, the expression is given as

$$\boldsymbol{J}_M[i] = \left.\frac{\partial p(E|M,\Theta)}{\partial \theta\{\frac{M}{\boldsymbol{m}_i}\}}\right|_{\hat{\Theta}}$$

Note that because values of Θ are not variable in the conditions stated by Eqn (4.1) and (4.2) (these non-varible conditions exist whenever $\boldsymbol{m}_i \not\supset m$), the following derivatives have zero value

$$\left.\frac{\partial p(E|M,\Theta)}{\partial \theta\{\frac{M}{\boldsymbol{m}_i}\}}\right|_{\hat{\Theta}} = 0 \qquad \forall\ \boldsymbol{m}_i \not\supset m$$

The expressions for the partial derivatives with respect to $p(E|M,\Theta)$ are obtained by differentiating Eqn(4.5). For compactness of notation, we introduce $\boldsymbol{S}$ and $\boldsymbol{n}$ and $\hat{\boldsymbol{\theta}}$ as vectors.

$$\begin{aligned}
\boldsymbol{S} &= [S(E|\boldsymbol{m}_1), S(E|\boldsymbol{m}_2), \ldots, S(E|\boldsymbol{m}_n)] \\
\boldsymbol{n} &= [n(\boldsymbol{m}_1), n(\boldsymbol{m}_2), \ldots, n(\boldsymbol{m}_n)] \\
\hat{\boldsymbol{\theta}} &= \left[\hat{\theta}\{\tfrac{M}{\boldsymbol{m}_1}\}, \hat{\theta}\{\tfrac{M}{\boldsymbol{m}_2}\}, \ldots, \hat{\theta}\{\tfrac{M}{\boldsymbol{m}_n}\}\right]
\end{aligned}$$

For non-zero conditions, the partial differentials for the Jacobian are then given as

$$\left.\frac{\partial p(E|M,\Theta)}{\partial \theta\{\frac{M}{\boldsymbol{m}_i}\}}\right|_{\hat{\Theta}} = \frac{\boldsymbol{n}_i \boldsymbol{S}_i}{\sum_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}} - \frac{\boldsymbol{n}_i \ \sum_k \boldsymbol{S}_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k}{\left(\sum_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k\right)^2}$$

Terms for the Hessian are given as

$$\boldsymbol{H}_M[i,j] = \left.\frac{\partial^2 p(E|M,\Theta)}{\partial\theta\{\frac{M}{\boldsymbol{m}_i}\}\,\partial\theta\{\frac{M}{\boldsymbol{m}_j}\}}\right|_{\hat{\Theta}}$$

with similar zero-derivative conditions

$$\left.\frac{\partial^2 p(E|M,\Theta)}{\partial\theta\{\frac{M}{\boldsymbol{m}_i}\}\,\partial\theta\{\frac{M}{\boldsymbol{m}_j}\}}\right|_{\hat{\Theta}} = 0 \qquad \forall \begin{array}{l} \boldsymbol{m}_i \not\supset M \\ \boldsymbol{m}_j \not\supset M \end{array}$$

For non-zero conditions, the second-order partial differentials for the Hessian are given as

$$\left.\frac{\partial^2 p(E|M,\Theta)}{\partial\theta\{\frac{M}{\boldsymbol{m}_i}\}\,\partial\theta\{\frac{M}{\boldsymbol{m}_j}\}}\right|_{\hat{\Theta}} = -\frac{\boldsymbol{n}_i\boldsymbol{S}_j + \boldsymbol{n}_j\boldsymbol{S}_i}{\left(\sum_k \boldsymbol{n}_k\hat{\boldsymbol{\theta}}_k\right)^2} + \frac{\boldsymbol{n}_i\boldsymbol{n}_j\ \sum_k \boldsymbol{S}_k\boldsymbol{n}_k\hat{\boldsymbol{\theta}}_k}{\left(\sum_k \boldsymbol{n}_k\hat{\boldsymbol{\theta}}_k\right)^3}$$

With terms for the Jacobian and Hessian already defined, the resulting second order expression is

$$p(E|M,\Theta) = \hat{p}(E|M) + \boldsymbol{J}_M(\Theta - \hat{\Theta}) + \frac{1}{2}(\Theta - \hat{\Theta})^T\boldsymbol{H}_M(\Theta - \hat{\Theta}) \tag{4.12}$$

where $\hat{p}(E|M)$ is the informed likelihood estimate

$$\hat{p}(E|M) = p(E|M,\hat{\Theta})$$

4.4.3 The second-order Bayesian combination rule

After the second-order approximation has been obtained for the likelihood, it can be combined with priors. Priors may also contain associated ambiguity (especially in the case of dynamic application) so we consider a general case where both the priors and the likelihoods are represented by second-order expressions.

$$\begin{aligned} p(E|M,\Theta) &= \hat{p}(E|M) + \boldsymbol{J}_{(E|M)}(\hat{\Theta} - \Theta) + \frac{1}{2}(\hat{\Theta} - \Theta)^T\boldsymbol{H}_{(E|M)}(\hat{\Theta} - \Theta) \\ p(M|\Theta) &= \hat{p}(M) + \boldsymbol{J}_{(M)}(\hat{\Theta} - \Theta) + \frac{1}{2}(\hat{\Theta} - \Theta)^T\boldsymbol{H}_{(M)}(\hat{\Theta} - \Theta) \\ p(E) &= \sum_m \hat{p}(E|M)\hat{p}(M) \end{aligned}$$

Bayesian combination is performed by taking the following product:

$$p(M|E,\Theta) = \frac{1}{p(E)}p(E|M,\Theta)p(M|\Theta)$$

By collecting terms with respect to $(\hat{\Theta} - \Theta)$, the posterior probability is expressed as

$$p(M|E,\Theta) = \hat{p}(M|E) + \boldsymbol{J}_{(M|E)}(\hat{\Theta} - \Theta) + \frac{1}{2}(\hat{\Theta} - \Theta)^T\boldsymbol{H}_{(M|E)}(\hat{\Theta} - \Theta) \tag{4.13}$$

where the terms $\hat{p}(M|E), \boldsymbol{J}_{(M|E)}, \boldsymbol{H}_{(M|E)}$ are calculated as

$$\hat{p}(M|E) = \frac{1}{p(E)}\hat{p}(E|M)\hat{p}(M) \tag{4.14}$$

$$\boldsymbol{J}_{(M|E)} = \frac{1}{p(E)}\left[\boldsymbol{J}_{(M)}\hat{p}(E|M) + \boldsymbol{J}_{(E|M)}\hat{p}(M)\right] \tag{4.15}$$

$$\begin{aligned}\boldsymbol{H}_{(M|E)} = \frac{1}{p(E)}\big[&\boldsymbol{H}_{(M)}\hat{p}(E|M) + \boldsymbol{H}_{(E|M)}\hat{p}(M) + \\ &\boldsymbol{J}^T_{(M)}\boldsymbol{J}_{(E|M)} + \boldsymbol{J}^T_{(E|M)}\boldsymbol{J}_{(M)}\big]\end{aligned} \tag{4.16}$$

These expressions form the second-order update rules. In addition these rules can be used to combine independent likelihoods

$$p(E|M,\Theta) = p(E_1|M,\Theta)p(E_2|M,\Theta)$$

However, there is no normalization constant $p(E)$, thus the second-order rules for combining independent evidence are

$$\hat{p}(E|M) = \hat{p}(E_1|M)\hat{p}(E_2|M) \tag{4.17}$$

$$\boldsymbol{J}_{(E|M)} = \left[\boldsymbol{J}_{(E_1|M)}\hat{p}(E_2|M) + \boldsymbol{J}_{(M|E_1)}\hat{p}(M|E_2)\right] \tag{4.18}$$

$$\begin{aligned}\boldsymbol{H}_{(E|M)} = \big[&\boldsymbol{H}_{(E_1|M)}\hat{p}(E_2|M) + \boldsymbol{H}_{(M|E_1)}\hat{p}(M|E_2) + \\ &\boldsymbol{J}^T_{(M|E_1)}\boldsymbol{J}_{(M|E_2)} + \boldsymbol{J}^T_{(M|E_1)}\boldsymbol{J}_{(M|E_2)}\big]\end{aligned} \tag{4.19}$$

4.5 Brief Comparison of Combination Methods

An example comparing the Fagin-Halpern and Second-Order boundaries is borrowed from [52]. In this example, evidence is available from six conditionally independent sources for a seven-mode system. When a new piece of evidence was available, the sources of evidence yielded the support as given in Table 4.1. These six sources of evidence are combined with the following prior:

$$[p(m_1), \ldots, p(m_7)] = [0.27\ ,\ 0.09\ ,\ 0.09\ ,\ 0.19\ ,\ 0.11\ ,\ 0.15\ ,\ 0.11]$$

Combination was done using the exact method (which resulted in a very complicated function), the second-order method, and the Fagin-Halpern method. The posterior probability results are shown in Figure (4.1). Three probabilities are shown, the plausibility (indicated by the lightest bar), the informed probability (indicated by the mid-coloured bar) and the belief (indicated by the darkest bar). From what can be seen from six combinations, the Fagin-Halpern probability ranges are extremely large, as was expected, with a range of practically 0%–100% for all modes. Such a result is definitively inconclusive for diagnosis. By contrast, the second-order method yields probability boundaries that are much closer to the true result.

Table 4.1: Support from example scenario

Mode	Support $s(E\|M)$						# of Observations
	e_1	e_2	e_3	e_4	e_5	e_6	
$\{1\}$	0.29	0.15	0.17	0.17	0.20	0.40	164
$\{2\}$	0.14	0.25	0.15	0.15	0.15	0.25	82
$\{3\}$	0.30	0.20	0.32	0.22	0.10	0.30	79
$\{4\}$	0.20	0.20	0.23	0.59	0.22	0.22	89
$\{5\}$	0.23	0.19	0.17	0.20	0.36	0.21	64
$\{6\}$	0.17	0.24	0.14	0.14	0.13	0.23	62
$\{7\}$	0.09	0.08	0.11	0.11	0.14	0.34	103
$\{1,2\}$	0.18	0.21	0.20	0.16	0.16	0.34	31
$\{1,3\}$	0.29	0.21	0.39	0.18	0.15	0.40	31
$\{1,4\}$	0.29	0.24	0.19	0.39	0.22	0.31	32
$\{1,5\}$	0.27	0.16	0.21	0.30	0.34	0.44	29
$\{1,6\}$	0.20	0.22	0.15	0.14	0.22	0.27	29
$\{1,7\}$	0.16	0.31	0.28	0.14	0.16	0.36	34
$\{2,6\}$	0.22	0.21	0.19	0.22	0.26	0.24	18
$\{3,4\}$	0.25	0.23	0.27	0.34	0.16	0.27	21
$\{4,5\}$	0.21	0.19	0.25	0.43	0.27	0.31	20
$\{5,6\}$	0.29	0.20	0.15	0.23	0.23	0.24	16
$\{4,5,6\}$	0.19	0.19	0.26	0.32	0.30	0.22	18

4.6 Applying the Second-Arder Rule Dynamically

4.6.1 Unambiguous dynamic solution

One of the strongest motivations for using the second-order combination rule is its ability to express ambiguity, even after successive combination. Successive combination is *heavily used* when solutions are applied in a dynamic manner to take into account autodependent modes. The dynamic solution to the autodependent mode problem has been discussed in Chapter 2, but it was only applicable to the case where no modes had ambiguity. However, the second-order probability expression in Eqn (4.13) has been formulated in such a manner that would enable easy application in a dynamic setting.

From the fundamentals, it was noted that the probability transition solution was

$$p(M^t|E^{t-1}) = \sum_{i=1}^{n} p(M^t|m_i^{t-1})p(m_i^{t-1}|E^{t-1}) \tag{4.20}$$

where $p(m_i^{t-1}|E^{t-1})$ is the posterior probability of mode m_i at time $t-1$. The probability transition rule calculates a prior probability at time t ($p(M^t|E^{t-1})$) using the posterior probability from $t-1$ ($p(m_i^{t-1}|E^{t-1})$). This resultant prior $p(M^t|E^{t-1})$ is used as the prior probability, to calculate the posterior at time t using Bayes' Theorem

$$p(m_i^t|E^t) = \frac{p(E^t|M)p(M^t|E^{t-1})}{\sum_M p(E^t|M)p(M^t|E^{t-1})} \tag{4.21}$$

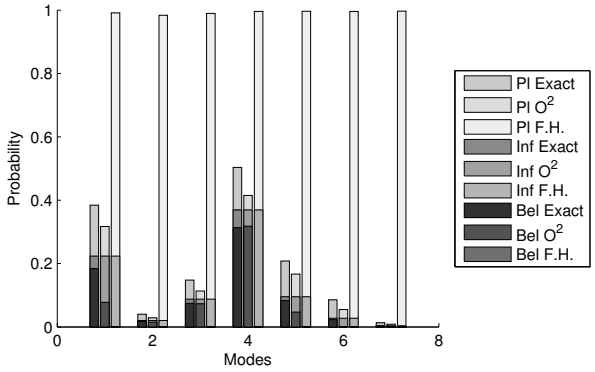


Figure 4.1: Diagnosis result for support in Table 4.1

4.6.2 The second-order dynamic solution

In this chapter, the second-order version of Bayes' theorem has already been derived. The remaining work is to define the *second-order probability transition rule*. Consider a second-order posterior probability that was obtained at time $t-1$

$$p(M^{t-1}|E^{t-1},\Theta) = \hat{p}(M^{t-1}|E^{t-1}) + \boldsymbol{J}_{(M^{t-1}|E^{t-1})}(\hat{\Theta}-\Theta) + \frac{1}{2}(\hat{\Theta}-\Theta)^T \boldsymbol{H}_{(M^{t-1}|E^{t-1})}(\hat{\Theta}-\Theta) \tag{4.22}$$

The second-order prior probability at time t can be obtained by directly applying Eqn (4.22) to Eqn (4.20)

$$p(M^t|E^{t-1},\Theta) = \sum_{i=1}^{n} p(M^t|m_i^{t-1}) p(m_i^{t-1}|E^{t-1},\Theta)$$

If we consider each second-order term for $p(M^{t-1}|E^{t-1},\Theta)$ in Eqn (4.22), a second order transition rule can be made for each term when transitioning to M^t.

$$\hat{p}(M^t|E^{t-1}) = \sum_{i=1}^{n} p(M^t|m_i^{t-1}) \hat{p}(M^{t-1}|E^{t-1}) \tag{4.23}$$

$$\boldsymbol{J}_{(M^t|E^{t-1})} = \sum_{i=1}^{n} p(M^t|m_i^{t-1}) \boldsymbol{J}_{(M^{t-1}|E^{t-1})} \tag{4.24}$$

$$\boldsymbol{H}_{(M^t|E^{t-1})} = \sum_{i=1}^{n} p(M^t|m_i^{t-1}) \boldsymbol{H}_{(M^{t-1}|E^{t-1})} \tag{4.25}$$

This resulting probability is used as a prior probability at time t, which can be updated

to a posterior using the previously proposed second-order Bayesian combination rule:

$$\begin{aligned}
\hat{p}(E^t) &= \sum_M \hat{p}(E^t|M)\hat{p}(M^t|E^{t-1}) \\
\hat{p}(M^t|E^t) &= \frac{1}{\hat{p}(E^t)}\hat{p}(E^t|M)\hat{p}(M^t|E^{t-1}) \\
\boldsymbol{J}_{(M|E)} &= \frac{1}{\hat{p}(E^t)}\left[\boldsymbol{J}_{(M^t|E^{t-1})}\hat{p}(E^t|M) + \boldsymbol{J}_{(E^t|M)}\hat{p}(M^t|E^{t-1})\right] \\
\boldsymbol{H}_{(M|E)} &= \frac{1}{\hat{p}(E^t)}\Big[\boldsymbol{H}_{(M^t|E^{t-1})}\hat{p}(E^t|M) + \boldsymbol{H}_{(E^t|M)}\hat{p}(M^t|E^{t-1}) + \\
&\qquad \boldsymbol{J}^T_{(M^t|E^{t-1})}\boldsymbol{J}_{(E^t|M)} + \boldsymbol{J}^T_{(E^t|M)}\boldsymbol{J}_{(M^t|E^{t-1})}\Big]
\end{aligned}$$

4.7 Making a Diagnosis

After using the second-order combination rule to merge prior probabilities and likelihoods, the results can be used for diagnosis. The second-order approximation is convenient as it can be used to define four quantities useful for diagnosis:

1. The informed probability $\hat{p}(M|E)$
2. The belief $Bel(M|E)$
3. The plausibility $Pl(M|E)$
4. The expected probability $E_\Theta[p(M|E,\Theta)]$

4.7.1 Simple diagnosis

The first quantity $\hat{p}(M|E)$ can be used as a simple diagnosis reference. This is a convenient quantity to use because it is explicitly available in the second-order result. In fact, if one simply wishes to obtain a simple diagnosis, $\hat{p}(M|E)$ is the only term that needs to be calculated; $\boldsymbol{J}_{(M|E)}$ and $\boldsymbol{H}_{(M|E)}$ are not needed. One simply chooses the mode which has the largest informed posterior $\hat{p}(M|E)$.

4.7.2 Ranged diagnosis

It may be desirable to also convey information about the ambiguity associated with a diagnosis. For example, if evidence is located in a region where historical data tends to be ambiguous, the probability range will be large. Conversely in regions where historical data tends to be unambiguous, the probability range will be small. Probability ranges can be calculated according to

$$\begin{aligned}
Bel(M|E) &= \min_\Theta p(M|E,\Theta) \\
Pl(M|E) &= \max_\Theta p(M|E,\Theta)
\end{aligned}$$

Because the expression is quadratic, the resulting minimization and maximization problems can be solved using quadratic programming methods; hence converting the expression to a second-order approximation greatly simplifies the minimization and maximization procedures. Note that the following constraints must be applied:

$$0 \leq \theta\{\tfrac{M}{\boldsymbol{m}_k}\} \leq 1$$

While the probability boundaries are approximate, the function $p(M|E,\Theta)$ is generally well-approximated by the second-order approximation over its domain (limited between 0 and 1 for all Θ). The probability boundaries serve to give an estimate on how reliable the diagnosis is, and how adversely it is affected by ambiguity in the historical data.

4.7.3 Expected value diagnosis

One can also use the second-order approximation to obtain an expected value $E_\Theta[p(M|E,\Theta)]$. However, in doing this, one is required to treat Θ as a random variable and construct a probability distribution for it. Due to the fact that elements in Θ can be seen as probabilities themselves

$$\theta\{\tfrac{M}{\boldsymbol{m}_k}\} = p(M|\boldsymbol{m}_k)$$

an appropriate distribution for Θ is the Dirichlet distribution, a probability distribution often used to define the distribution of probability estimates.

The Dirichlet Distribution for expected values of $p(M|E,\Theta)$

Let us consider $\Theta\{\frac{\bullet}{\boldsymbol{m}_k}\}$, the set of elements in Θ that pertain to the ambiguous mode $\boldsymbol{m}_k$

$$\Theta\{\tfrac{\bullet}{\boldsymbol{m}_k}\} = \left[\theta\{\tfrac{m_1}{\boldsymbol{m}_k}\}, \ldots, \theta\{\tfrac{m_n}{\boldsymbol{m}_k}\}\right]$$

These elements behave like a complete set of discrete probabilities, which follow the Dirichlet distribution

$$\begin{aligned} f(\ \Theta\{\tfrac{\bullet}{\boldsymbol{m}_k}\} \mid \boldsymbol{\alpha}\{\tfrac{\bullet}{\boldsymbol{m}_k}\}\ ) &= \frac{\Gamma\left(\sum_i \alpha\{\frac{m_i}{\boldsymbol{m}_k}\}\right)}{\prod_i \Gamma(\alpha\{\frac{m_i}{\boldsymbol{m}_k}\})} \prod_i \theta\{\tfrac{m_i}{\boldsymbol{m}_k}\}^{\alpha\{\frac{m_i}{\boldsymbol{m}_k}\}-1} \\ &= \frac{1}{B(\alpha)} \prod_i \theta\{\tfrac{m_i}{\boldsymbol{m}_k}\}^{\alpha\{\frac{m_i}{\boldsymbol{m}_k}\}-1} \\ \boldsymbol{\alpha}\{\tfrac{\bullet}{\boldsymbol{m}_k}\} &= \left[\alpha\{\tfrac{m_1}{\boldsymbol{m}_k}\}, \ldots, \alpha\{\tfrac{m_n}{\boldsymbol{m}_k}\}\right] \end{aligned}$$

where $B(\alpha)$ is a normalization constant. As previously mentioned, the values $\Theta\{\frac{\bullet}{\boldsymbol{m}_k}\}$, can be seen as probabilities and must be on the interval between 0 and 1; furthermore, they must sum to unity. The terms $\boldsymbol{\alpha}\{\frac{\bullet}{\boldsymbol{m}_k}\}$ are shape parameters that can be interpreted as the

number of prior samples. Thus the Dirichlet distribution is a probability distribution of probability estimates given the samples. The expected value of $\theta\{\frac{m_i}{\boldsymbol{m}_k}\}$ is given as

$$E(\theta\{\tfrac{m_i}{\boldsymbol{m}_k}\}) = \frac{\alpha\{\frac{m_i}{\boldsymbol{m}_k}\}}{\sum \boldsymbol{\alpha}\{\frac{\bullet}{\boldsymbol{m}_k}\}} \tag{4.26}$$

As an example to aid in the interpretation of a Dirichlet distribution, let us assign the following values to the parameters:

$$\theta\{\tfrac{m_i}{\boldsymbol{m}_k}\} = p(m_i|\boldsymbol{m}_k)$$
$$\boldsymbol{\alpha}\{\tfrac{m_i}{\boldsymbol{m}_k}\} = n(m_i|\boldsymbol{m}_k)$$

where $p(m_i|\boldsymbol{m}_k)$ is the probability of mode m_i given ambiguous mode $\boldsymbol{m}_k$ and $n(m_i|\boldsymbol{m}_k)$ is the frequency of mode m_i given ambiguous mode $\boldsymbol{m}_k$. The Dirichlet distribution is given as

$$f(\ p(M|\boldsymbol{m}_k) \mid n(M|\boldsymbol{m}_k)\ ) = \frac{\Gamma\left(\sum_i n(m_i|\boldsymbol{m}_k)\right)}{\prod_i \Gamma(n(m_i|\boldsymbol{m}_k))} \prod_i p(m_i|\boldsymbol{m}_k)^{n(m_i|\boldsymbol{m}_k)-1}$$

Using this Dirichlet distribution, the expected value of $\theta\{\frac{m_i}{\boldsymbol{m}_k}\}$ can be calculated as

$$E[p(m_i|\boldsymbol{m}_k)] = \frac{n(m_i|\boldsymbol{m}_k)}{\sum_j n(m_j|\boldsymbol{m}_k)} = \frac{n(m_i|\boldsymbol{m}_k)}{n(\boldsymbol{m}_k)}$$

This is the probability one would expect to obtain given the samples $n(m_j|\boldsymbol{m}_k)$. Note that the expected values are always in the interval of 0 and 1, and sum to unity (given $\boldsymbol{m}_k$) as is required.

The previous Dirichlet distribution was used to denote the PDF $f(\Theta\{\frac{\bullet}{\boldsymbol{m}_k}\})$ for Θ parameters pertaining to a single ambiguous mode. Now we would like to express a PDF $f(\Theta)$ pertaining to all ambiguous modes. Note that parameter sets for different ambiguous modes are independent of each other, resulting in the following expression for $f(\Theta)$

$$f(\Theta) = \prod_k f(\Theta\{\tfrac{\bullet}{\boldsymbol{m}_k}\})$$

This probability distribution defines the possible values of Θ and will be used for calculating the expected value.

Calculating the expected value of $p(M|E,\Theta)$

The posterior probability $p(M|E,\Theta)$ was previously given in Eqn (4.13); by making use of the probability distribution over Θ in $f(\Theta)$, the expected value is given as

$$\begin{aligned} E[p(M|E)] &= \int_\Theta \left[\hat{p}(M|E) + \boldsymbol{J}_{(M|E)}\Delta\Theta + \frac{1}{2}\Delta\Theta^T\boldsymbol{H}_{(M|E)}\Delta\Theta\right] \prod_k f(\Theta\{\tfrac{\bullet}{\boldsymbol{m}_k}\})\ d\Theta \\ &= \text{Const} + \int_\Theta \left[\boldsymbol{J}^*_{(M|E)}\Theta + \frac{1}{2}\Theta^T\boldsymbol{H}_{(M|E)}\Theta\right] \prod_k f(\Theta\{\tfrac{\bullet}{\boldsymbol{m}_k}\})\ d\Theta \end{aligned} \tag{4.27}$$

where

$$\text{Const} = \hat{p}(M|E) - \boldsymbol{J}_{(M|E)}\hat{\Theta} + \frac{1}{2}\hat{\Theta}^T \boldsymbol{H}_{(M|E)}\hat{\Theta}$$
$$\boldsymbol{J}^*_{(M|E)} = (\boldsymbol{J}_{(M|E)} - \hat{\Theta}^T \boldsymbol{H}_{(M|E)})$$

The second-order expression of $p(M|E,\Theta)$ is linear with respect to Θ and contain terms no higher than second-order. Because of this, the expected value can be expressed in terms of means and variances, which have well-known solutions for the Dirichlet distribution.

$$\int_\Theta \theta\{\tfrac{M}{\boldsymbol{m}_i}\} \prod_k f(\Theta\{\tfrac{\bullet}{\boldsymbol{m}_k}\})\, d\Theta = E(\theta\{\tfrac{M}{\boldsymbol{m}_i}\})$$
$$\int_\Theta \theta\{\tfrac{M}{\boldsymbol{m}_i}\}\theta\{\tfrac{M}{\boldsymbol{m}_j}\} \prod_k f(\Theta\{\tfrac{\bullet}{\boldsymbol{m}_k}\})\, d\Theta = E(\theta\{\tfrac{M}{\boldsymbol{m}_i}\})E(\theta\{\tfrac{M}{\boldsymbol{m}_i}\})$$
$$\int_\Theta \theta^2\{\tfrac{M}{\boldsymbol{m}_i}\} \prod_k f(\Theta\{\tfrac{\bullet}{\boldsymbol{m}_k}\})\, d\Theta = \left[E(\theta\{\tfrac{M}{\boldsymbol{m}_i}\})\right]^2 + \text{Var}(\theta\{\tfrac{M}{\boldsymbol{m}_i}\})$$

For the Dirichlet distribution, the means and variances are given by

$$A(\boldsymbol{m}_i) = \sum_{m_k \subset \boldsymbol{m}_i} \alpha(m_k|\boldsymbol{m}_i)$$
$$E(\theta\{\tfrac{M}{\boldsymbol{m}_i}\}) = \frac{\alpha(M)}{A(\boldsymbol{m}_i)}$$
$$\text{Var}(\theta\{\tfrac{M}{\boldsymbol{m}_i}\}) = \frac{\alpha(M)[A(\boldsymbol{m}_i) - \alpha(M)]}{A(\boldsymbol{m}_i)^3 + A(\boldsymbol{m}_i)^2}$$

where $\alpha(M)$ represents the prior sample of the unambiguous mode M. If one uses the frequency of mode occurrences $n(M)$, the shape parameters $\alpha(M)$ will be quite large and the expected value of $p(M|E,\Theta)$ will be nearly identical to the informed estimate. This is because a large shape parameters $\alpha(M)$ yield a very sharp distribution centred at $\hat{\Theta}$. If one is unsure about the accuracy of $\hat{\Theta}$ it is best to divide all $\alpha(M)$ by a common factor so that the largest $\alpha(M)$ is no larger than 10 (values larger than 10 can result in fairly narrow distributions).

The means and variances can be applied to Eqn (4.27). First, one has to separate the squared terms of Θ associated with variance

$$E[p(E|M)] = \text{Const} + \int_\Theta \left[\boldsymbol{J}^*_{(M|E)}\Theta + \frac{1}{2}\left[\ \Theta^T(\boldsymbol{H}_{(M|E)} - \boldsymbol{H_D})\Theta + [\Theta^2]^T(\boldsymbol{H_D}\ \mathbf{1})\ \right]\right] f(\Theta)\, d\Theta$$

where $\boldsymbol{H_D}$ is a diagonal matrix containing the diagonal elements of $\boldsymbol{H}_{(M|E)}$, and $\mathbf{1}$ is a vertical vector of ones. Furthermore, Θ is a vertical vector of parameters, and Θ^2 is a vertical vector containing squared values of Θ. By expressing the integrals as expected values and variances, the solution is reduced to

$$E[p(E|M)] = \text{Const} + \boldsymbol{J}^*_{(M|E)}E(\Theta) + \frac{1}{2}E(\Theta)^T(\boldsymbol{H}_{(M|E)} - \boldsymbol{H_D})E(\Theta) + \frac{1}{2}[\text{Var}(\Theta) + E(\Theta)^2]^T \boldsymbol{H_D}\ \mathbf{1} \tag{4.28}$$

This can be solved in order to obtain the expected value of the posterior.

When $E_\Theta[p(M|E,\Theta)]$ approaches $\hat{p}(M|E)$

As was mentioned earlier, if the number of prior samples $n(M)$ is large, one can justify using the informed estimate $\hat{p}(E|M)$ by assuming strong prior knowledge. In such a case, the prior samples α are large so that $\alpha \to \infty$. The expected values and variances will then take the following values:

$$\begin{aligned} E(\theta\{\tfrac{M}{\boldsymbol{m}_i}\}) &= \hat{\theta}\{\tfrac{M}{\boldsymbol{m}_i}\} \\ \mathrm{Var}(\theta\{\tfrac{M}{\boldsymbol{m}_i}\}) &= 0 \end{aligned}$$

If this occurs, it can be shown that Eqn (4.27) will revert to the informed estimate $\hat{p}(M|E)$.

$$\begin{aligned} E[p(M|E)] &= \mathrm{Const} + \boldsymbol{J}^*_{(M|E)}\hat{\Theta} + \frac{1}{2}\left[\hat{\Theta}^T(\boldsymbol{H}_{(M|E)} - \boldsymbol{H_D})\hat{\Theta} + [\mathbf{0} + \hat{\Theta}^2]^T\boldsymbol{H_D}\ \mathbf{1}\right] \\ &= \mathrm{Const} + [\boldsymbol{J}_{(M|E)} - \hat{\Theta}^T\boldsymbol{H}_{(M|E)}]\hat{\Theta} \\ &\quad + \frac{1}{2}\left[\hat{\Theta}^T(\boldsymbol{H}_{(M|E)} - \boldsymbol{H_D})\hat{\Theta} + \hat{\Theta}^T\boldsymbol{H_D}\hat{\Theta}\right] \\ &= \mathrm{Const} + \left[\boldsymbol{J}_{(M|E)}\hat{\Theta} - \hat{\Theta}^T\boldsymbol{H}_{(M|E)}\hat{\Theta}\right] + \frac{1}{2}\hat{\Theta}^T\boldsymbol{H}_{(M|E)}\hat{\Theta} \\ &= \left[\hat{p}(M|E) - \boldsymbol{J}_{(M|E)}\hat{\Theta} + \frac{1}{2}\hat{\Theta}^T\boldsymbol{H}_{(M|E)}\hat{\Theta}\right] + \boldsymbol{J}_{(M|E)}\hat{\Theta} - \frac{1}{2}\hat{\Theta}^T\boldsymbol{H}_{(M|E)}\hat{\Theta} \\ &= \hat{p}(M|E) \end{aligned}$$

This result justifies using $\hat{p}(M|E)$ when the prior sample size is large, or equivalently, if one has high confidence in the priors. However, if one is not confident in the value of $\hat{\Theta}$, (or equivalently, if one is not confident that prior probabilities adequately represent proportions in the ambiguous modes) one should use small sample sizes of α and calculate the expected value for diagnosis.

Chapter 5

Accounting for Ambiguous Modes: A Dempster-Shafer Approach

5.1 Introduction

Inference methods for ambiguous hypotheses have existed since the late sixties with Dempster [49] and Shafer [50] being the first to contribute major publications in this field; a field which later became known as Dempster-Shafer theory. First, being proposed as a generalization to Bayesian inference, Dempster-Shafer theory was shown to be able to account for both probabilistic uncertainty and ignorance, the claim being that Bayesian inference cannot adequately express ignorance. Since its inception, there has been a vast amount of literature published on Dempster-Shafer theory, which includes a wide arrangement of combination rules, methods of interpretation, and criticisms, (criticisms are mainly due to the subjective nature of Dempster-Shafer theory).

A solution to the diagnosis problem with ambiguous modes has been proposed in Chapter 4 using the parametrized Bayesian method; however, Dempster-Shafer theory provides an alternative solution. It may seem that Demspter-Shafer theory can be readily applied to the ambiguous mode problem, but further investigation reveals some difficult challenges that ultimately requires restructuring and generalization of some of the basic concepts of Dempster-Shafer theory. Nevertheless, because of the intended scope of Dempster-Shafer theory is quite broad, the method in this chapter does not require certain assumptions to be made about $\boldsymbol{\Theta}$ as in Chapter 4.

5.2 Dempster-Shafer Theory

5.2.1 Basic belief assignments

The principal difference between Dempster-Shafer theory and Bayesian inference is the interpretation of probability. In the Bayesian sense, all of the supported hypotheses must be mutually exclusive. For example, let us borrow the problem in Chapter 4 with a valve

and a sensor, with each of the two components having three states.

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \\ m_7 \\ m_8 \\ m_9 \end{bmatrix} = \begin{bmatrix} c_1 = 1\ ,\ c_2 = 1 \\ c_1 = 1\ ,\ c_2 = 2 \\ c_1 = 1\ ,\ c_2 = 3 \\ c_1 = 2\ ,\ c_2 = 1 \\ c_1 = 2\ ,\ c_2 = 2 \\ c_1 = 2\ ,\ c_2 = 3 \\ c_1 = 3\ ,\ c_2 = 1 \\ c_1 = 3\ ,\ c_2 = 2 \\ c_1 = 3\ ,\ c_2 = 3 \end{bmatrix}$$

All modes in this case, are exclusive hypotheses. When there is ambiguity in any of these modes, overlap between the hypotheses will exist. Probability can only be applied to a set of exclusive hypotheses, which in our case is an unambiguous mode m.

$$p(M) = \frac{n(M)}{n}$$

Dempster-Shafer theory however, makes use of a Basic Belief Assignment (BBA) which is equivalent to the support function defined Chapter 4.

$$S(\boldsymbol{M}) = \frac{n(\boldsymbol{M})}{n}$$
$$1 = \sum_{\boldsymbol{M}} S(\boldsymbol{M})$$

where $\boldsymbol{M}$ can contain ambiguous modes (for example, $\boldsymbol{m}_k = \{m_1, m_2, m_3\}$ which occurs when $c_1 = 1$ is observed and c_2 is missing); when ambiguous modes are in the data, overlapping hypotheses can occur. Dempster-Shafer theory aims to express the probability in terms of the BBA. When invoking the Θ parameter notation given in Chapter 4, the probability is expressed as

$$p(\boldsymbol{M}|\Theta) = \sum_{\boldsymbol{m}_k \cap \boldsymbol{M} \neq \emptyset} \theta\{\tfrac{\boldsymbol{M}}{\boldsymbol{m}_k}\} S(\boldsymbol{m}_k)$$

where $\boldsymbol{M}$ is the mode of interest, and $\theta\{\frac{\boldsymbol{M}}{\boldsymbol{m}_k}\}$ is an unknown proportion parameter that represnts the probability of $\boldsymbol{M}$ given $\boldsymbol{m}_k$.

$$\theta\{\tfrac{\boldsymbol{M}}{\boldsymbol{m}_k}\} = p(\boldsymbol{M}|\boldsymbol{m}_k)$$

The parameters $\theta\{\frac{\boldsymbol{M}}{\boldsymbol{m}_k}\}$ have a set of constraints. The first constraint

$$0 \leq \theta\{\tfrac{\boldsymbol{M}}{\boldsymbol{m}_k}\} \leq 1$$

states that, because $\theta\{\frac{\boldsymbol{M}}{\boldsymbol{m}_k}\}$ is a probability, it cannot be larger than 1, or smaller than 0. In two special cases, $\theta\{\frac{\boldsymbol{M}}{\boldsymbol{m}_k}\}$ is not random at all, but must take on specific values based

on logical constraints. In the first case, if the mode of interest $\boldsymbol{M}$ completely contains the mode $\boldsymbol{m}_k$, then all the support to $\boldsymbol{m}_k$ must apply to $\boldsymbol{M}$.

$$\theta\{\tfrac{\boldsymbol{M}}{\boldsymbol{m}_k}\} = 1 \qquad \forall \;\; \boldsymbol{M} \supseteq \boldsymbol{m}_k \tag{5.1}$$

For example if the mode of interest $\boldsymbol{M}$ is the ambiguous mode $\{m_1, m_2, m_3\}$ and the supported mode is $\{m_1, m_3\}$ then all support given to $\{m_1, m_3\}$ in the history must also be given to $\{m_1, m_2, m_3\}$.

In the second case, if the mode of interest $\boldsymbol{M}$ has nothing in common with the mode $\boldsymbol{m}_k$, none of the support given to $\boldsymbol{m}_k$ can apply to $\boldsymbol{M}$.

$$\theta\{\tfrac{\boldsymbol{M}}{\boldsymbol{m}_k}\} = 0 \qquad \forall \;\; \boldsymbol{M} \cap \boldsymbol{m}_k = \emptyset \tag{5.2}$$

Outside of these conditions, $\theta\{\frac{\boldsymbol{M}}{\boldsymbol{m}_k}\}$ is a flexible (or unknown) value between 0 and 1.

$$0 \leq \theta\{\tfrac{\boldsymbol{M}}{\boldsymbol{m}_k}\} \leq 1 \qquad \forall \;\; \boldsymbol{M} \not\supseteq \boldsymbol{m}_k \;\; , \;\; \boldsymbol{M} \cap \boldsymbol{m}_k = \emptyset \tag{5.3}$$

5.2.2 Probability boundaries

Dempster-Shafer theory concerns itself with boundaries on the probability. The plausibility and belief can be obtained by maximizing and minimizing $p(\boldsymbol{M}|\Theta)$ over the unknown parameters Θ. The optimization in this problem is linear with respect to Θ and is constrained by previously mentioned conditions. Because the BBA $S(\boldsymbol{m}_k)$ values (which serve as coefficients on Θ) are non-negative, the belief can be obtained by setting all flexible values of Θ to zero.

$$Bel(\boldsymbol{M}) = Ex(\boldsymbol{M}) = \sum_{\boldsymbol{m}_k \subseteq \boldsymbol{M}} S(\boldsymbol{m}_k)$$

Because the condition $\boldsymbol{m}_k \supseteq \boldsymbol{M}$ excludes support from all flexible values of Θ, it is called the *exclusive condition* $Ex(\boldsymbol{M})$. For Dempster-Shafer theory, the exclusive probability is the solution to the belief or lower bound probability.

In a similar manner, the plausibility can be obtained by setting all flexible values of Θ to 1.

$$Pl(\boldsymbol{M}) = In(\boldsymbol{M}) = \sum_{\boldsymbol{m}_k \cap \boldsymbol{M} \neq \emptyset} S(\boldsymbol{m}_k)$$

The condition $\boldsymbol{m}_k \cap \boldsymbol{M} \neq \emptyset$ includes support from all flexible values of Θ, and so is called the *inclusive condition* $In(\boldsymbol{M})$. In this way, the inclusive probability is the solution to the plausibility or upper bound probability.

5.2.3 Dempster's rule of combination

Dempter's Rule of combination is made to combine two BBAs of $\boldsymbol{M}$ from independent sources, and is said to be a generalization of Bayeisan combination. Dempter's rule can be

expressed as

$$S_{1,2}(\boldsymbol{M}) = \frac{1}{1-K} \sum_{\boldsymbol{M}=\boldsymbol{m}_i \cap \boldsymbol{m}_j \neq \emptyset} S_1(\boldsymbol{m}_i) S_2(\boldsymbol{m}_j)$$
$$K = \sum_{\emptyset=\boldsymbol{m}_i \cap \boldsymbol{m}_j} S_1(\boldsymbol{m}_i) S_2(\boldsymbol{m}_j)$$

Here, to find out the combined support $S_{1,2}(\boldsymbol{M})$ of the mode $\boldsymbol{M}$ we search for all modes in S_1 and S_2 that intersect to yield $\boldsymbol{M}$ (expressed as $\boldsymbol{M} = \boldsymbol{m}_i \cap \boldsymbol{m}_j \neq \emptyset$). However, there is no such thing as support to the empty set $\emptyset$ which denotes conflict. Support to conflict is denoted as K, and is normalized out (as $1-K$), because BBAs are not allowed to support conflict.

In Dempster-Shafer theory, a BBA is called Bayesian if it contains no support for ambiguous hypotheses. In our application, the BBA is Bayesian if support is only given to unambiguous modes. If BBAs are Bayesian, then Dempster's Rule will revert to Bayes' Theorem.

$$S_{1,2}(M) = \frac{1}{1-K} \sum_{m=m_i \cap m_j \neq \emptyset} S_1(m_i) S_2(m_j) = \frac{1}{1-K} S_1(M) S_2(M)$$
$$K = \sum_{\emptyset=m_i \cap m_j} S_1(\boldsymbol{m}_i) S_2(\boldsymbol{m}_j) = 1 - \sum_m S_1(M) S_2(M)$$

so that the end result is

$$S_{1,2}(M) = \frac{1}{\sum_m S_1(M) S_2(M)} S_1(M) S_2(M)$$

which indeed resembles Bayes' Theorem.

Dempster's Rule will always yield support to intersections between $S_1(\boldsymbol{m}_i)$ and $S_2(\boldsymbol{m}_j)$. Because of this, ambiguity is reduced after every combination; the idea is that information from $S_1(\boldsymbol{m}_i)$ will be applied to ambiguity in $S_2(\boldsymbol{m}_j)$ and information from $S_2(\boldsymbol{m}_j)$ will be applied to ambiguity in $S_1(\boldsymbol{m}_i)$. Applying information to each others' ambiguity will reduce the uncertainty between the two BBAs.

Because each combination results in a reduction in ambiguity, the more combinations that are performed, the more that the resulting BBA will resemble a Bayesian BBA. In fact, if any BBA is combined with a Bayesian BBA, the resulting BBA will be Bayesian. In such a case, precise information from the Bayesian BBA will be applied to the uncertainties in the other BBAs, resulting in zero uncertainty after combination.

5.2.4 Shortcut combination for unambiguous priors

If the prior probability is Bayesian, Dempster-Shafer combination of any BBA will yield a Bayesian result. In such a case, a short-cut solution is available to calculate the posterior which is much less computationally intensive than applying Dempster's rule directly. In

addition, when implementing a dynamic solution, successive combinations yield a Bayesian result, thus it makes sense to use a Bayesian prior in order to cut computational loads. Choosing Bayesian priors not only reduces computational loads, it also yields a dynamic application that is fully compatible with the dynamic Bayesian method; the posteriors are always Bayesian, thus probability transition technique will be the same as in the dynamic Bayesian solution.

Let us consider a BBA $S_1(M)$ that is a Bayesian prior $p_1(M)$, and another BBA $S_2(\boldsymbol{M})$ which contains ambiguity. Using Dempster's rule for combination

$$S_{12}(M) = \frac{1}{1-K} \sum_{M=m_i \cap \boldsymbol{m}_j \neq \emptyset} p_1(m_i) S_2(\boldsymbol{m}_j)$$

Now the condition

$$M = m_i \cap \boldsymbol{m}_j \neq \emptyset$$

is only true when $m_i = M$ and when $\boldsymbol{m}_j \supseteq M$. Because of this, we can factor out $p_1(M)$ and replace the condition with $\boldsymbol{m}_j \supseteq m$.

$$S_{12}(M) = \frac{1}{1-K} p_1(M) \sum_{\boldsymbol{m}_j \supseteq m} S_2(\boldsymbol{m}_j)$$

Now because M is unambiguous, the condition $\boldsymbol{m}_j \supseteq M$ is equivalent to $\boldsymbol{m}_j \cap M \neq \emptyset$ so that

$$S_{12}(M) = \frac{1}{1-K} p_1(M) \sum_{\boldsymbol{m}_j \cap M \neq \emptyset} S_2(\boldsymbol{m}_j)$$

From earlier results, we can see that the S_2 term amounts to the *inclusive probability* of M (or equivalently, the Dempster-Shafer plausibility).

$$S_{12}(M) = \frac{1}{1-K} p_1(M) In_2(M) \tag{5.4}$$

Because $1-K$ is a normalization constant (so that $\sum_M S_{12}(M) = 1$) we can define $1-K$ as

$$1 - K = \sum_M p_1(M) In_2(M) \tag{5.5}$$

Eqn (5.4) and (5.5) together define the short-cut evaluation of Dempster's rule with a Bayesian prior.

Dynamic application

When using the shortcut rule, the posterior result is always Bayesian, thus the transition-rule is still the same

$$S^t(M) = \sum_k p(M^t | m_k^{t-1}) S^{t-1}(m_k)$$

After Dempster's rule was applied at a time step $t-1$, the resulting probability $S^{t-1}(m_k)$ can be converted to the prior $S^t(M)$ at time t using the transition rule. Desmpter's combination rule is then used to update $S^t(M)$ with more evidence.

5.3 Generalizing Dempster-Shafer Theory

In the previously discussed Fagin-Halpern combination rule, boundaries grow after each combination, and in the second-order Bayesian method boundaries tend to stay relatively constant; however in the Dempster-Shafer method, boundaries tend to shrink after successive combinations. The reason for this is that Dempster's rule does not make the assumption that Θ values are identical but that they are independent and that information from one BBA can make up for the ambiguity in another. As a rule for application, if reference data from different information sources is taken from the same time window, it is best to use the second-order Bayesian method, because Θ values will be identical. However, if the evidence data for each source comes from different time intervals, it is better to apply Generalized Dempster-Shafer theory as Θ values will be independent.

Applying Dempster-Shafer theory to our problem does not come without difficulties, as will be seen later on in this section; the BBA does not adequately describe how ambiguity affects the likelihood. Because of this, the BBA and Dempster's rule need to be generalized in order to better fit the problem in question.

Previously, when discussing Dempster-Shafer theory, we concerned ourselves about the probabilities of all modes in the history $p(\boldsymbol{M})$; however, we are only interested in diagnosing unambiguous modes $p(M)$. From this point forward, we will be only considering the problem of diagnosing unambiguous modes $p(M)$ with potentially ambiguous modes $\boldsymbol{M}$ in the history.

5.3.1 Motivation: Difficulties with BBAs

The difficulty with using Dempster-Shafer theory is representing the likelihood as a BBA. Demspter-Shafer theory can be used to describe direct probabilities with ambiguity.

$$p(M|E,\Theta) = \sum_{\boldsymbol{m}_k \cap m \neq \emptyset} \theta\{\tfrac{M}{\boldsymbol{m}_k}\} S(\boldsymbol{m}_k|E) \tag{5.6}$$

where the BBA terms in S are given as

$$S(\boldsymbol{M}|E) = \frac{n(\boldsymbol{M},E)}{n(E)}$$

This is the case where we sample data at random from the entire evidence history, $\mathcal{D}$, and we assume that the mode frequencies in $\mathcal{D}$ represent the mode probabilities for the population. When evaluating the probability directly, we consider the number of times the mode M and evidence E occur simultaneously ($n(\boldsymbol{M},E)$) and divide by the total number of times the evidence E occurs ($n(E)$).

The disadvantage for direct evaluation is that in most cases, some modes occur quite rarely, thus we cannot trust that $\mathcal{D}$ is representative of the mode frequency. Furthermore, dynamic application of the direct method is difficult. Instead it may be better to obtain the priors using process knowledge (from both this process, and possibly other similar processes) and use Bayes' Theorem to combine likelihoods from the data with these priors. The results of utilizing such an approach are increased flexibility in sampling from $\mathcal{D}$, and the ability for easy dynamic implementation.

Evaluating likelihoods however, poses a problem for Dempster-Shafer theory; the likelihood with respect to S and Θ is given in Chapter 4 as

$$p(E|M,\Theta) = \frac{\sum\limits_{\boldsymbol{m}_k \supseteq M} \theta\{\frac{M}{\boldsymbol{m}_k}\} S(E|\boldsymbol{m}_k) n(\boldsymbol{m}_k)}{\sum\limits_{\boldsymbol{m}_k \supseteq m} \theta\{\frac{M}{\boldsymbol{m}_k}\} n(\boldsymbol{m}_k)} \tag{5.7}$$

which is very different from Eqn (5.6). There are in fact, two main functional differences between the Dempster-Shafer problem in Eqn (5.6) and our problem in Eqn (5.7).

Difference 1

In the Dempster-Shafer problem, the term $S(\boldsymbol{m}_k|E)$ functions as a *non-negative* coefficient on $\theta\{\frac{M}{\boldsymbol{m}_k}\}$ which means that increasing $\theta\{\frac{M}{\boldsymbol{m}_k}\}$ never decreases $p(M|E,\Theta)$

$$\frac{\partial\ p(M|E,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m}_k}\}} \geq 0 \tag{5.8}$$

In our problem, because we have a fractional expression now, it is possible for an increase in $\theta\{\frac{M}{\boldsymbol{m}_k}\}$ to result in a decrease in $p(E|M,\boldsymbol{\theta}\{m\})$ (an example of this is shown later)

$$\frac{\partial\ p(E|M,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m}_k}\}} \ngeq 0 \tag{5.9}$$

Because derivatives are no longer non-negative, $Bel(E|M)$ is not necessarily solved by using the exclusive probability, and $Pl(E|M)$ is no longer solved by using the inclusive probability.

Difference 2

In the Dempster-Shafer problem, the term $S(\boldsymbol{m}_k|E)$ functions as a *constant* coefficient on $\theta\{\frac{M}{\boldsymbol{m}_k}\}$ with respect to m. This means that as long as m_i and m_j are both in $\boldsymbol{m}_k$, the partial derivatives of $p(m_i|E,\Theta), p(m_j|E,\Theta)$ are the same with respect to their θ parameters on m.

$$\frac{\partial\ p(m_i|E,\Theta)}{\partial\ \theta\{\frac{m_i}{\boldsymbol{m}_k}\}} = \frac{\partial\ p(m_j|E,\Theta)}{\partial\ \theta\{\frac{m_j}{\boldsymbol{m}_k}\}} \qquad m_i, m_j \subset \boldsymbol{m}_k \tag{5.10}$$

In our problem, because of the fractional expression, the normalization constant on the denominator can change with respect to M, thus the partial derivatives of $p(E|m_i,\Theta), p(E|m_j,\Theta)$

are not necessarily the same with respect to their θ parameters on m.

$$\frac{\partial\ p(E|m_i,\Theta)}{\partial\ \theta\{\frac{m_i}{\boldsymbol{m}_k}\}} \neq \frac{\partial\ p(E|m_j,\Theta)}{\partial\ \theta\{\frac{m_j}{\boldsymbol{m}_k}\}} \qquad m_i, m_j \subset \boldsymbol{m}_k \tag{5.11}$$

Example of differences

Let us consider a simple two-mode system with one ambiguous mode so that

- $\boldsymbol{m}_1 = m_1$
- $\boldsymbol{m}_2 = m_2$
- $\boldsymbol{m}_3 = \{m_1, m_2\}$

The historical data for this system is presented in Table 5.1.

Table 5.1: Frequency counts from example

	m_1	m_2	$\{m_1, m_2\}$	All $\boldsymbol{M}$
e_1	12	4	7	23
e_2	5	9	6	20
All e	17	13	13	43

Now let us consider a case where e_1 is observed. We can see that directly evaluating the probability of m yields

$$p(m_1|e_1,\Theta) = \frac{12}{23} + \theta\{\tfrac{m_1}{m_1,m_2}\}\frac{7}{23}$$
$$p(m_2|e_1,\Theta) = \frac{4}{23} + \theta\{\tfrac{m_2}{m_1,m_2}\}\frac{7}{23}$$

One can observe that the derivative of these expressions with respect to $\theta\{\frac{M}{m_1,m_2}\}$ is 7/23. This result is identical for both m_1, m_2 and is also non-negative. Conversely, when evaluating the likelihoods of E given the modes m_1 and m_2 we obtain

$$p(e_1|m_1,\Theta) = \frac{12 + \theta\{\frac{m_1}{m_1,m_2}\}7}{17 + \theta\{\frac{m_1}{m_1,m_2}\}13}$$
$$p(e_1|m_2,\Theta) = \frac{4 + \theta\{\frac{m_2}{m_1,m_2}\}7}{13 + \theta\{\frac{m_2}{m_1,m_2}\}13}$$

We can further see that the derivatives can be obtained as

$$\frac{\partial\ p(e_1|m_1,\Theta)}{\partial\ \theta\{\frac{m_1}{m_1,m_2}\}} = \frac{-37}{(17 + \theta\{\frac{m_1}{m_1,m_2}\}13)^2}$$
$$\frac{\partial\ p(e_1|m_2,\Theta)}{\partial\ \theta\{\frac{m_2}{m_1,m_2}\}} = \frac{39}{(13 + \theta\{\frac{m_2}{m_1,m_2}\}13)^2}$$

We can see that the derivative of $p(e_1|m_1,\Theta)$ is always negative, while the derivative of $p(e_1|m_2,\Theta)$ is always positive. Thus the derivatives for likelihoods are not identical nor are they non-negative.

Implications of the differences

The implication of these results is that we cannot adequately express the likelihood expression $p(E|M,\Theta)$ in Eqn (5.7) in the Dempster-Shafer BBA format presented in Eqn (5.6). A more general form of the BBA is required to express $p(E|M,\Theta)$, and a new combination rule must also be constructed.

5.3.2 Generalizing the BBA

At this point, we have established that the Dempster-Shafer BBAs make use of three assumptions which are not true for our Bayesian inference problem:

1. The function $p(E|M,\Theta)$ is linear with respect to Θ
2. The derivatives of $p(E|M,\Theta)$ are always positive with respect to Θ
3. The derivatives of $p(E|M,\Theta)$ with respect to $\theta\{\frac{M}{\boldsymbol{m}_k}\}$ are identical for all M if $\boldsymbol{m}_k$ is held constant

The objective of the Generalized BBA is to relax assumptions (2) and (3) in order to express $p(E|m_i,\Theta)$ as a first-order approximation of Θ

$$p(E|m,\Theta) = \boldsymbol{G}[:,m]^T \boldsymbol{\Theta}[:,m] \tag{5.12}$$

Here, $\boldsymbol{G}$ and $\boldsymbol{\Theta}$ take structures that allow us to easily define a generalized Dempster's rule of combination. In this thesis $\boldsymbol{G}[:,m]$ denotes the m^{th} column of $\boldsymbol{G}$, while $\boldsymbol{G}[m,:]$ denotes the m^{th} row of $\boldsymbol{G}$.

A first-order approximation is used in this chapter, firstly because Dempster's rule of Combination is a linear operation and difficult to generalize over higher orders. Secondly, combination operations tend to reduce ambiguity and thus tend to reduce the influence of Θ on the final result; this reduced influence of Θ results in posterior functions $p(M|E,\Theta)$ that are increasingly linear (and increasingly constant) as combinations are performed, depreciating the relevance of higher-order terms.

In this thesis, $\boldsymbol{\Theta}$ is the matrix form of Θ, with each row representing the potentially ambiguous mode $\boldsymbol{m}_k$, and each column representing an unambiguous mode m_i so that

$$\boldsymbol{\Theta}[k,i] = \theta\{\tfrac{m_i}{\boldsymbol{m}_k}\}$$

The matrix $\boldsymbol{G}$ has the same dimensions as $\boldsymbol{\Theta}$ where elements can be calculated as

$$\boldsymbol{G}[k,i] = \begin{cases} 0 & m_i \cap \boldsymbol{m}_k = \emptyset \\ \tilde{p}(E|m_i) & m_i = \boldsymbol{m}_k \\ \frac{\partial p(E|m_i)}{\partial \boldsymbol{\theta}[k,i]} & m_i \subset \boldsymbol{m}_k \end{cases} \tag{5.13}$$

where

$$\tilde{p}(E|m_i) = p(E|m_i, \hat{\Theta}) - \sum_{\boldsymbol{m}_k \supset m_i} \hat{\theta}\{\tfrac{m_i}{\boldsymbol{m}_k}\} \left. \frac{\partial\ p(E|m_i, \Theta)}{\partial\ \theta\{\tfrac{m_i}{\boldsymbol{m}_k}\}} \right|_{\hat{\Theta}}$$

$$\frac{\partial p(E|m_i)}{\partial \boldsymbol{\Theta}[k,i]} = \left. \frac{\partial\ p(E|m_i, \Theta)}{\partial\ \theta\{\tfrac{m_i}{\boldsymbol{m}_k}\}} \right|_{\hat{\Theta}}$$

Note that $\hat{\boldsymbol{\Theta}}$ is the reference value of $\boldsymbol{\Theta}$, around which the approximation is centered. In Chapter 4, $\hat{\boldsymbol{\Theta}}$ was defined as the *informed probability*. In this chapter, because of the properties of Dempster's rule, it is best to use the *inclusive value* of $\boldsymbol{\Theta}$.

$$\hat{\boldsymbol{\Theta}} = \boldsymbol{\Theta}^*$$

where

- $\boldsymbol{\Theta}^*$ is the *inclusive value*, which sets all *flexible* values to 1
- $\boldsymbol{\Theta}_*$ is the *exclusive value*, which sets all *flexible* values to 0

Note that the mathematical conditions for *flexible* values were given in Eqn (5.3). With $\boldsymbol{G}$ and $\boldsymbol{\Theta}$ defined in this manner, Eqn (5.12) is able to express the likelihood as a first-order approximation.

One important feature of $\boldsymbol{G}$ is that the mode can be extracted by the values taken on by the row of $\boldsymbol{G}$. For example, let us consider a three mode system, with possible ambiguous modes $\{m_1, m_2\}$, $\{m_1, m_3\}$, and $\{m_2, m_3\}$. The structure of $\boldsymbol{G}$ is defined as

$$\boldsymbol{G} = \left[\begin{array}{c|ccc} & m_1 & m_2 & m_3 \\ \hline m_1 & G\{\frac{m_1}{m_1}\} & 0 & 0 \\ m_2 & 0 & G\{\frac{m_2}{m_2}\} & 0 \\ m_3 & 0 & 0 & G\{\frac{m_3}{m_3}\} \\ \{m_1, m_2\} & G\{\frac{m_1}{m_1,m_2}\} & G\{\frac{m_2}{m_1,m_2}\} & 0 \\ \{m_1, m_3\} & G\{\frac{m_1}{m_1,m_3}\} & 0 & G\{\frac{m_3}{m_1,m_3}\} \\ \{m_2, m_3\} & 0 & G\{\frac{m_2}{m_2,m_3}\} & G\{\frac{m_3}{m_2,m_3}\} \end{array}\right]$$

We can therefore recover the mode from the appropriate row of $\boldsymbol{G}$ by analysing the zero elements. For example, the fourth row of $\boldsymbol{G}$ is

$$\boldsymbol{G}[4, :] = \left[\begin{array}{ccc} G\{\frac{m_1}{m_1,m_2}\} & G\{\frac{m_2}{m_1,m_2}\} & 0 \end{array}\right] \rightarrow \{m_1, m_2\}$$

where each column in this row pertains to modes m_1, m_2, m_3. Because the third element is zero, m_3 is not supported, and hence, $\{m_1, m_2\}$ is supported. Thus for every row of $\boldsymbol{G}$, zeros indicate that the corresponding modes are not supported. In a similar manner, we can see that the row

$$\boldsymbol{G}[2, :] = \left[\begin{array}{ccc} 0 & G\{\frac{m_2}{m_2}\} & 0 \end{array}\right] \rightarrow m_2$$

only supports m_2. In this way, we can see that the mode can be recovered by determining which elements in $\boldsymbol{G}[k, :]$ are equal to zero.

This generalized form of the BBA, because it is a generalization, can be used to express Dempster-Shafer BBAs as well. If we consider the conditions set forth by Eqn (5.8) and Eqn(5.10), and the GBBA construction method in Eqn (5.13) we can set about two conditions where the GBBA can be classified as a BBA.

1. Every non-zero element in $\boldsymbol{G}$ must be non-negative
2. Every non-zero element in a given row $\boldsymbol{G}[k, :]$ must have identical values

By taking the GBBA structure from our previous three-mode system, $\boldsymbol{G}$ would be a BBA if it took the form

$$\boldsymbol{G} = \left[\begin{array}{c|ccc} & m_1 & m_2 & m_3 \\ \hline m_1 & S(m_1) & 0 & 0 \\ m_2 & 0 & S(m_2) & 0 \\ m_3 & 0 & 0 & S(m_3) \\ \{m_1, m_2\} & S\{m_1, m_2\} & S\{m_1, m_2\} & 0 \\ \{m_1, m_3\} & S\{m_1, m_3\} & 0 & S\{m_1, m_3\} \\ \{m_2, m_3\} & 0 & S\{m_2, m_3\} & S\{m_2, m_3\} \end{array}\right]$$

which is the GBBA structure that would be formed if the direct method (as opposed to the Bayesian likelihood method) was used.

Probability boundaries on GBBAs

When GBBAs are applied, the inclusive and exclusive probabilities have similar definitions to BBAs

$$Ex(M) = \sum_{\boldsymbol{m}_k \subseteq m} \boldsymbol{G}[m_k, m] = \boldsymbol{G}[:, m]^T \boldsymbol{\Theta}_*[:, m] \tag{5.14}$$

$$In(M) = \sum_{\boldsymbol{m}_k \supseteq m} \boldsymbol{G}[m_k, m] = \boldsymbol{G}[:, m]^T \boldsymbol{\Theta}^*[:, m] \tag{5.15}$$

where $\boldsymbol{\Theta}^*[:, m]$ sets all flexible θ values to 1, while $\boldsymbol{\Theta}_*[:, m]$ sets all flexible θ values to 0. While the inclusive and exclusive probability definitions are similar, they are not the solutions to the belief and plausibility.

$$Bel(M) = \min_{\boldsymbol{\Theta}} \boldsymbol{G}[:, m]^T \boldsymbol{\Theta}[:, m] \neq Ex(M)$$

$$Pl(M) = \max_{\boldsymbol{\Theta}} \boldsymbol{G}[:, m]^T \boldsymbol{\Theta}[:, m] \neq In(M)$$

5.3.3 Generalizing Dempster's rule

As previously mentioned, Dempster's Rule of combination is given for BBAs in the following form:

$$S(\boldsymbol{m}_k) = \frac{1}{1-K} \sum_{\boldsymbol{m}_k = \boldsymbol{m}_i \cap \boldsymbol{m}_j \neq \emptyset} S(\boldsymbol{m}_i) S(\boldsymbol{m}_j) \tag{5.16}$$

$$1 - K = \sum_{\boldsymbol{m}_k = \boldsymbol{m}_i \cap \boldsymbol{m}_j \neq \emptyset} S(\boldsymbol{m}_i) S(\boldsymbol{m}_j) \tag{5.17}$$

In a similar manner, the Generalized Dempster's Rule of Combination is applied to the rows of $\boldsymbol{G}$ that pertain to $\boldsymbol{m}_k$ (or equivalently, $G[\boldsymbol{m}_k, :]$)

$$G_{12}[\boldsymbol{m}_k, :] = \frac{1}{1-K} \sum_{\boldsymbol{m}_k = \boldsymbol{m}_i \cap \boldsymbol{m}_j \neq \emptyset} G_1[\boldsymbol{m}_i, :] \circ G_2[\boldsymbol{m}_j, :] \tag{5.18}$$

$$1 - K = \sum_{\boldsymbol{m}_k = \boldsymbol{m}_i \cap \boldsymbol{m}_j \neq \emptyset} \text{mean}_{x \neq 0}(G[\boldsymbol{m}_i, :] \circ G[\boldsymbol{m}_j, :]) \tag{5.19}$$

where $X \circ Y$ denotes the Hadamard (or element-wise) product between X and Y, while $\text{mean}_{x \neq 0}(X)$ is the mean of the non-zero values of X.

The interesting property about the Hadamard product is that it conserves properties of the intersection $x \cap y$ operation. For example, we can see that

$$[\ 0 \quad X_2 \quad 0\ ] \circ [\ 0 \quad Y_2 \quad Y_3\ ] = [\ 0 \quad X_2 Y_2 \quad 0\ ]$$

Analogously, when obtaining sets from these row vectors, we can see that

$$m_2 \cap \{m_2, m_3\} = m_2$$

When taking this into account, we can see that the Generalized Dempster's Rule of combination truly generalizes Dempster's rule. For example, let us consider two BBA values for $\{m_1, m_2\}$ and $\{m_2, m_3\}$. In Dempster's rule, we could allocate the product

$$S\{m_1, m_2\} S\{m_2, m_3\}$$

to mode m_2. In the Generalized Dempster's rule, the Hadamard product

$$\begin{aligned}[S\{m_1, m_2\}, S\{m_1, m_2\},\ 0\ ] \circ [\ 0\ , S\{m_2, m_3\}, S\{m_2, m_3\}] \\ = [\ 0\ , S\{m_1, m_2\} S\{m_2, m_3\},\ 0\ ]\end{aligned}$$

is allocated to mode m_2 (which one can see from the product result). When the GBBA is consistent with a Dempster-Shafer BBA, the Hadamard product serves no purpose as all nonzero elements in the ambiguous mode $\boldsymbol{m}_k$ have the same support, regardless of the unambiguous modes $M \subset \boldsymbol{m}_k$ supported. The same final result can be obtained without Hadamard products as long as the sets and the BBA are known. However, when an ambiguous mode $\boldsymbol{m}_k$ in a GBBAs allocates different support to each unambiguous mode $M \subset \boldsymbol{m}_k$, the Hadamard product allows us keep track of how $\boldsymbol{m}_k$ supports each $M \subset \boldsymbol{m}_k$.

5.3.4 Shortcut combination for unambiguous priors

The Generalized Dempster's rule shares another property with Dempster's rule, that is, repeated combination will result in shrinking ambiguity, and that combination with a Bayesian prior will result in an unambiguous posterior. In fact, the Bayesian shortcut for the Generalized Dempster's rule is the same as the Bayesian shortcut for Dempster's rule. If $p_1(M)$ is a Bayesian prior, which is combined with a GBBA ($\boldsymbol{G}_2$), the resulting GBBA ($\boldsymbol{G}_{12}$) can be expressed as

$$\boldsymbol{G}_{12}(m_i, m_i) = \frac{1}{1-K} p_1(m_i) In_2(m_i) \tag{5.20}$$

$$\boldsymbol{G}_{12}(m_i, m_{j\neq i}) = 0$$

$$1 - K = \sum_M p_1(M) In_2(M) \tag{5.21}$$

This can be shown by analysing the Generalized rule of combination. However, we first need to define $p_1(M)$ as a GBBA

$$\boldsymbol{G}_1 = \begin{bmatrix} p_1(m_1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & p_1(m_n) \end{bmatrix}$$

From the Generalized rule,

$$G_{12}[m, :] = \frac{1}{1-K} \sum_{m=\boldsymbol{m}_i \cap \boldsymbol{m}_j \neq \emptyset} G_1[\boldsymbol{m}_i, :] \circ G_2[\boldsymbol{m}_j, :]$$

From this we can see that the condition $m = \boldsymbol{m}_i \cap \boldsymbol{m}_j \neq \emptyset$ is only satisfied when $\boldsymbol{m}_i = m$ and when $\boldsymbol{m}_j \supseteq m$. This allows us to factor out $G_1[m, :]$ which yields

$$G_{12}[m, :] = \frac{1}{1-K} G_1[m, :] \sum_{\boldsymbol{m}_j \supseteq m} G_2[\boldsymbol{m}_j, :]$$

We can see that for row 1 of G_1, the form is $[X, 0, \ldots, 0]$ which means that the first row of G_{12} will also take the form $[X, 0, \ldots, 0]$, where all elements other than the first one is zero. In the same manner, the second row of G_1 and $G_{1,2}$ will take the form of $[0, X, 0, \ldots, 0]$, so on and so forth. From this we can see that

$$G_{12}[m, m] = \frac{1}{1-K} G_1[m, m] \sum_{\boldsymbol{m}_j \supseteq m} G_2[\boldsymbol{m}_j, m]$$

$$G_{12}[m_i, m_{j\neq i}] = 0$$

Now, one can observe that the summation term is identical to that of inclusive probability described in Eqn (5.14), and that $G_1[m, m] = p_1(M)$. Thus,

$$G_{12}[m, m] = \frac{1}{1-K} p_1(M) In_2(M)$$

$$G_{12}[m_i, m_{j\neq i}] = 0$$

where the normalization constant assures that the diagonal sum of G_{12} is 1.

$$1 - K = \sum_{M} p_1(M) In_2(M)$$

This results provides a quick method to combine GBBAs with a Bayesian prior. In addition, the result validates the consistency of the Generalized Dempster's rule with Dempter's original rule.

Dynamic application

When using the shortcut rule, the posterior result is always Bayesian, thus the transition-rule is still the same

$$G^t(M, M) = \sum_{k} p(M^t | m_k^{t-1}) G^{t-1}(m_k, m_k)$$

After Dempster's rule was applied at a time step $t-1$, the resulting probability $G^{t-1}(M, M)$ can be converted to the prior $G^t(M, M)$ at time t using the transition rule. Desmpter's combination rule is then used to update $G^t(M, M)$ with more evidence.

Chapter 6

Making Use of Continuous Evidence Through Kernel Density Estimation

6.1 Introduction

When discussing material in Chapters 4 and 5, it was assumed that evidence was discretized in order to construct alarms. However, in the case of process monitors, the raw evidence is often continuous and discretization can result in the loss of valuable information. It is shown in this chapter that for a single monitor, discretization can be optimized to yield exactly the same result as continuous methods, but in higher dimensions, optimal discretization can be a challenge as the optimal regions can take on strange shapes.

Often times, parametric methods are used to estimate continuous distributions. However, parametric methods make assumptions about distribution shape. In the case of process monitors, the distributions can take very unusual shapes (for example, under certain situations, the control performance monitor results from the FCOR algorithm [34] will have bimodal behaviour with peaks near 0 and 1). The Gaussian Mixture Model approach is parametric but uses multiple Gaussian distributions to approximate the data distribution. The challenge for the Gaussian mixture model is that one is required to know beforehand how many Gaussian distributions are required, and estimation algorithms (such as EM) are not guaranteed to converge to the globally optimal solution. Furthermore, while Gaussian mixture distributions model multi-modal distributions quite well, there is still difficulty with distributions exhibiting non-linear behaviour between variables.

This chapter discusses kernel density estimation, a density estimation technique for continuous variables that is non-parametric. The main advantage of kernel density estimation is that it naturally follows the shape of the data and can adequately model the distributions regardless of the shape taken. Furthermore, the process of obtaining a kernel density estimate is not iterative, but is obtained in a single step; because of this, kernel density estimates yield the same consistent result for a single data set. However, much like the dis-

crete method, the kernel density method suffers from the curse of dimensionality in which performance will degrade in higher dimensions. Nevertheless, the performance degrades at a slower rate for kernel density methods.

This chapter discusses many important aspects of kernel density estimation, including performance relative to discrete methods, how to obtain the critically important bandwidth parameter and how to reduce dimensionality.

6.2 Performance: Continuous Methods vs. Discrete Methods

In previous chapters, discrete evidence was used mainly because of its ease of interpretation and the fact that any distribution can be discretized to estimate the probability distribution. Discretization is a non-parametric method in the sense that it does not require knowledge of the distribtion shape in order to approximate it (however, discrete distributions are categorical and have parametric properties such as the ability to perform Bayesian parameter updating). The drawback of discretization however, is that it results in a loss of information. By contrast kernel density estimation is a method that is applicable to continuous distributions and does not suffer as much from loss of information by discretization. It is also non-parametric, able to fit any type of distribution, which was the one key advantage to discrete methods. The comparison between discrete and kernel density methods is shown in Table (6.1)

Table 6.1: Comparison between kernel and discrete methods

Discrete	Kernel Density
Computationally light (when implemented intelligently)	Computationally heavy (proportional to data)
Suffers from information loss	Does not suffer from information loss
Performance suffers exponentially with increased dimensionality	Performance suffers exponentially with increased dimensionality (but much slower than the discrete case)

The advantages of the kernel density approach over the discrete approach are intuitive as there is much less loss of information. Mathematically, it has been shown that kernel density estimation converges to the true probability density function faster than discretization [73]. However, most readers may not be merely concerned about the accuracy of the density function estimate, but are more concerned about the false diagnosis rate. The material that

follows in this section will help readers understand the diagnosis performance characteristics of both methods.

6.2.1 Average false negative diagnosis criterion

In order to assess diagnostic performance, we must first choose a valid performance criterion. In this chapter, we calculate the *average false negative diagnosis rate* F_N as our performance criterion.

Our criterion takes note of every instance of a false diagnosis. In this chapter, the event that mode M diagnosed (or *chosen*) is denoted as $\mathcal{C}(M)$. From this, the events of true and false diagnosis can be expressed as

- $\mathcal{C}(M)|M$ represents a (true) diagnosis of M, when M is the true underlying mode.
- $\mathcal{C}(\bar{M})|M$ represents a (false) diagnosis of $\bar{M}$ when M is the true underlying mode (where $\bar{M}$ represents some mode other than M).

The *average false diagnosis rate* can be calculated as

$$F_N = \sum_{k=1}^{n_m} p(\mathcal{C}(\bar{m}_k)|m_k)p(m_k)$$

where $p(m_k)$ is the prior probability of m_k (a flat prior could be used if the prior probabilities are not known). When using Bayesian methods (which account for prior probability) the diagnosis is based on the maximum posterior probability.

$$\mathcal{C}(m_k) \quad \texttt{if} \quad m_k = \arg\max_M p(E|M)p(M)$$

The posterior probability density function is a likelihood function of E (i.e. $p(E|M)$) weighted by the prior probability $p(M)$. For certain regions of E, the mode M will be diagnosed (ideally, this is when E closely resembles data in M). Our notation of this region of E is given as

$$E|\mathcal{C}(M) = \text{The region of } E \text{ where } M \text{ is diagnosed}$$

Because E is continuous, we obtain $p(\mathcal{C}(\bar{M})|M)$ by integration over the regions of E where *M is not diagnosed*.

$$p(\mathcal{C}(\bar{M})|M) = \int_{E|\mathcal{C}(\bar{M})} p(E|M)p(M)\ de$$

This formal definition of F_N requires integration, which can be quite difficult to perform. In practice, the value of F_N is much easier to estimate using Monte Carlo simulations. This approach can be performed by taking the following steps:

1. Use training data to construct likelihoods $p(E|M)$.

2. Obtain validation data in proportion to the prior probabilities (for example, if the probability of m_1 is 40%, then 40% of the data must come from m_1).

3. Go through each of the data points E^t (taking note of its true mode M^t), and evaluate the posterior probability

$$p(M|E^t) = \frac{p(E^t|M)p(M)}{p(E^t)}$$

4. Find the mode with the maximum posterior probability, and set it as the diagnosed mode $\mathcal{C}(M)$.

5. In this step, we tally the diagnosis results, thus if E^t is correctly diagnosed we add one to the tally of correct results n_{cor}

$$n_{cor} = n_{cor} + 1 \qquad \texttt{if} \quad \mathcal{C}(M) = M^t$$

   Otherwise, if E^t is incorrectly diagnosed, we add one to the tally of incorrect results n_{inc}

$$n_{inc} = n_{inc} + 1 \qquad \texttt{if} \quad \mathcal{C}(M) \neq M^t$$

6. After tallying results for all E^t, the false negative rate F_N can be obtained using a simple quotient

$$F_N = \frac{n_{inc}}{n_{inc} + n_{cor}}$$

6.2.2 Performance of discrete methods vs continuous methods

When evaluating performance of methods between continuous and discretized evidence, one will arrive at two conclusions:

1. Continuous methods perform better than discrete methods, unless discrete methods are optimized in terms of boundary selection, at which point their performance is equal.

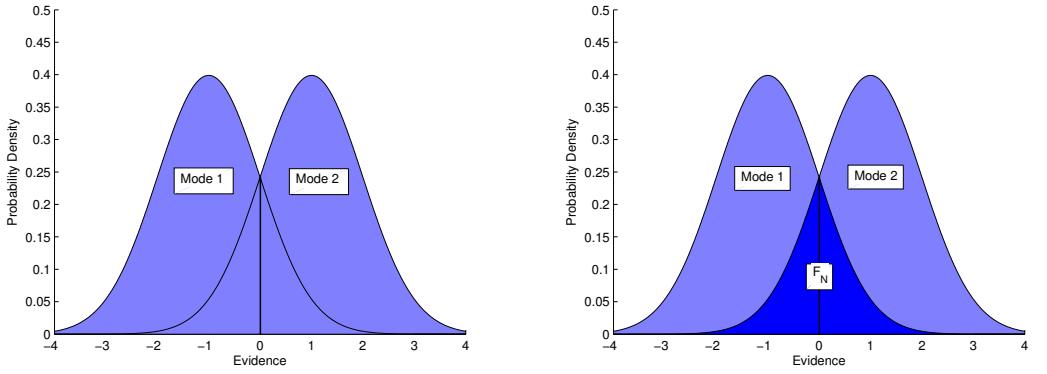
2. In a one-dimensional system, the optimal discretization boundary is easy to obtain, but in higher dimensions, defining the optimal discretization boundary can be difficult, if not infeasible.

When discrete methods perform as well as continuous

The discrete method performs as well as the continuous method when the discretization regions coincide with $E|\mathcal{C}(M)$

$$E|\mathcal{C}(M) = \text{The region of } E \text{ where } M \text{ is diagnosed}$$

As an example of defining optimal cutoff boundaries, let us consider a two-mode system where each mode is Gaussian, having the distributions given in Figure (6.1(a)). When continuous methods are used, we diagnose the mode that has the highest probability; thus if Mode 1 has a higher density function than Mode 2, Mode 1 is diagnosed. From this example, we can see that the diagnosis region for $E|\mathcal{C}(m_1)$ is where $E < 0$ and the region for $E|\mathcal{C}(m_2)$ is $E \geq 0$. Let us consider a case where $E < 0$, even if Mode 1 is more probable, it may not be the true mode because Mode 2 has non-zero probability in this region. In Figure (6.1), we see the overlapped regions shaded in a darker area, and these regions represent the false negative rates F_N. If this region is small, we have a low probability of misdiagnosis.



(a) Original Distributions

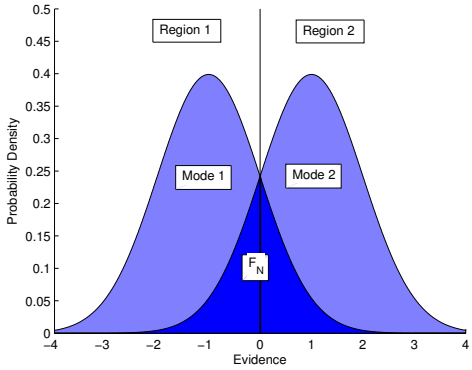
(b) Misdiagnosis rate of continuous method

Figure 6.1: Grouping approaches for kernel density method

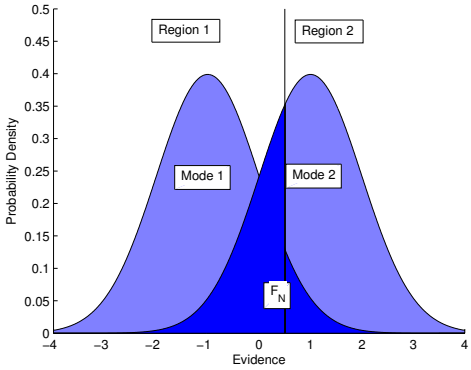
If we consider the discrete case, we must create discretization boundaries to analyse the probability distribution. The best boundaries for discretization are the ones set by the continuous distribution, in this example, we have one discrete region where $E < 0$ and one discrete region where $E \geq 0$. The resulting false negative rate is shown in Figure 6.2(a) which is identical to the continuous false diagnosis rate.

Now, let us consider the case where we have no knowledge of the continuous distributions. In this case we set the discretization regions to a new arbitrary location, where one region occurs at $E < 0.5$ and another region occurs at $E \geq 0.5$ as shown in Figure 6.2(b). Values of E to the left of this boundary will diagnose m_1, and values of E to the right of this boundary will diagnose m_2. When the boundary is shifted to this location, we can see that the shaded region representing F_N has somewhat grown. This is because when shifting this boundary from 0 to 0.5, the probability of falsely diagnosing Mode 2 (the right part of the dark region) decreased, but at a slower rate than the probability of falsely diagnosing Mode 1 (the left part of the dark region) grew.

From this example we can see that the optimal discretization region is given by the continuous distributions, and that shifting this region in any way will result in an increase in false diagnosis rates. Discrete methods can perform as well as continuous methods, but



(a) Misdiagnosis rate of optimal discrete



(b) Misdiagnosis rate of suboptimal discrete

Figure 6.2: Discrete method performance

optimal discretization requires knowledge of the continuous distributions, which begs the question as to why one would discretize the evidence in the first place. A short answer to this question is that the motivation for discretization is not for performance but for computational simplicity.

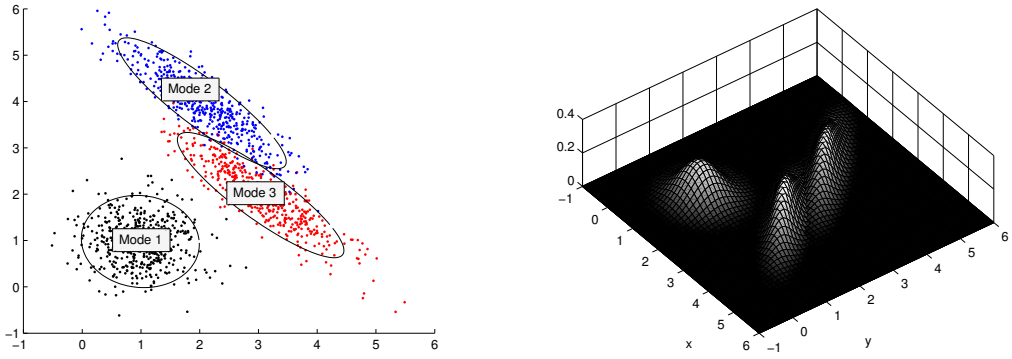
In addition to comparing performance between discrete and continuous cases, this example illustrates a useful procedure for optimal discretization; that is, to analyse the continuous distributions (either defined parametrically or through kernel density estimation), and note the regions where the continuous density for a target mode M is highest. This approach however, is only straightforward in one-dimensional cases because discretization boundaries can be more complex in higher dimensions.

The feasibility of optimal discrete methods in higher dimensions

In cases of dependent evidence E in dimensions two or higher, optimal discretization becomes a much more complicated endeavour than in our previous example, as discretization regions become more difficult to define (as they can be non-linear).

As an example on the difficulty of discretization, consider the two-dimensional three-mode system in Figure 6.3. In this case, we have three Gaussian distributions, and two of them are quite correlated (this type of behaviour can occur quite easily in practice).

The challenge with this system is that it is easiest to define discretization boundaries one at a time for each piece of evidence (this is called the element-wise approach). In this way, the discretization boundaries will be linear and follow the direction of the axes as seen in Figure 6.4(a). However, as one can see, this discretization scheme is quite suboptimal. When boundaries are drawn in this manner, region 1 will diagnose Mode 1, regions 2 and 4 will diagnose Mode 2, and region 3 will diagnose Mode 3. This scheme will yield a false negative rate of $F_N \approx 0.1233$. However, one can see that in this figure, the modes are actually quite well separated (the kernel density method has a false negative rate of



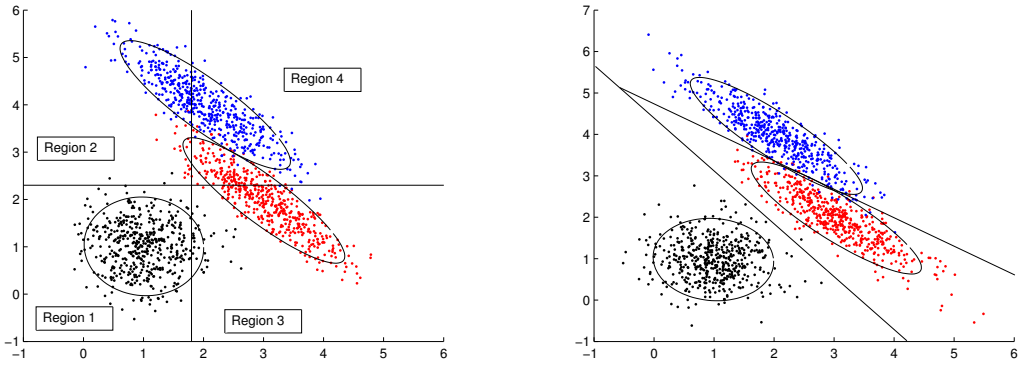
(a) Data from 2D system

(b) Distribution of 2D system

Figure 6.3: Two-dimensional system with dependent evidence

$F_N = 0.0020$).

If one will allow a more complex method to define the boundary, one can use visual inspection to draw linear boundaries as done in Figure 6.4(b). In such a case, the false diagnosis rate will be much closer to optimal $F_N = 0.0025$. However, optimally defining linear boundaries in higher dimensions becomes a very difficult task, especially when data takes on non-linear shapes.



(a) Element-wise discretization

(b) Nearly optimal linear discretization

Figure 6.4: Two-dimensional discretization schemes

6.3 Kernel Density Estimation

6.3.1 From histograms to kernel density estimates

As previously mentioned, kernel density estimation is a non-parametric method used to estimate continuous probability density functions. The process of arriving at a kernel density estimate from discrete frequency data is an intuitive one. First, let us consider a histogram

visualization of a distribution in Figure 6.5 which is synonymous to discretization. In this case, we divide the x axis into discrete segments, and we count the frequency of data points residing in each bin.

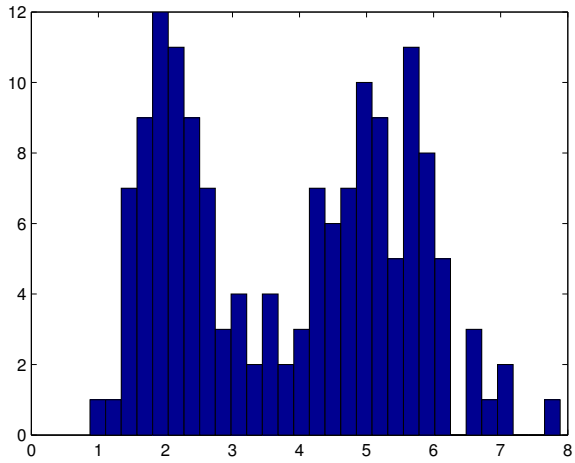


Figure 6.5: Histogram of distribution

In the histogram, we divide x into discrete segments, but what if we allowed the segments to be centred around each data point? This would mean that centred around each data point, we would place a rectangular function (with area $1/(n \times$ bin width), so that the distribution integrates to unity). After summing these rectangular functions, the end result (the centred histogram) will be slightly smoother than the histogram as seen in Figure 6.6.

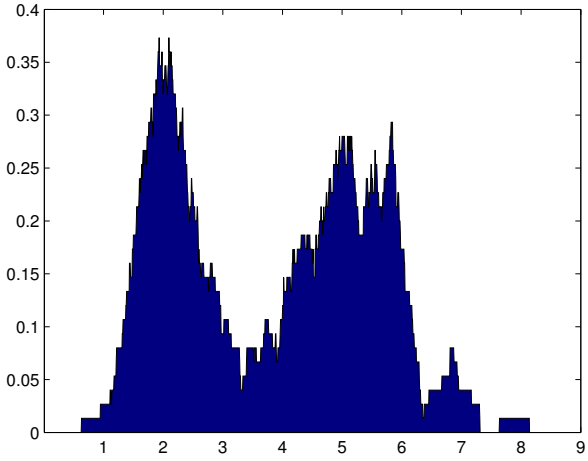


Figure 6.6: Centered histogram of distribution

The centred histogram is actually a kernel density estimation. The kernel in his case, is a block, with area $1/(n \times$ width). Instead of using this kernel, we could choose a smoother

kernel, for example, a Gaussian density function divided by n. This would yield the result in Figure 6.7.

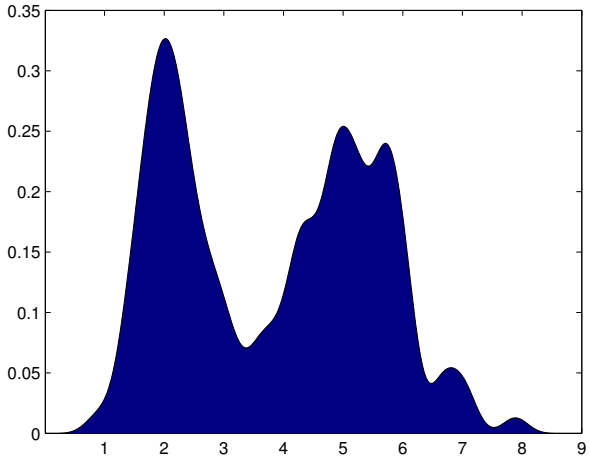


Figure 6.7: Gaussian kernel density estimate

6.3.2 Defining a kernel density estimate

From the example above, a kernel density estimate takes a set of sampled data points and places a kernel function, centred around each data point. As shown in Figure 6.8, the kernel functions (denoted by the blue dotted line), centred around five data points (denoted by the black dots on the x axis) will sum up to yield a smooth function (denoted by the solid black curve).

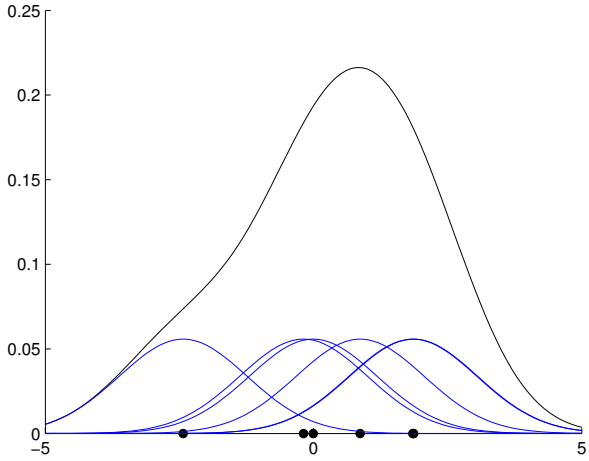


Figure 6.8: Kernels summing to a kernel density estimate

Mathematically, the kernel density estimation procedure can be defined as

$$f(x) \approx \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|H|^{1/2}} K\left(H^{1/2}(x - D_i)\right) \tag{6.1}$$

where D denotes a multivariate data set with n entries, $K(x)$ represents the kernel function, and H represents the bandwidth, which will be discussed in more detail later on. The kernel function itself can take on many forms as long as it is non-negative and the integral over the entire domain is equal to one.

$$\int_{-\infty}^{\infty} K(x)\, dx = 1$$

The two most popular kernels are

1. The Epanechnikov kernel (due to its asymptotic efficiency)
2. The standard multivariate normal kernel (due to its excellent differentiation properties and its ease of application in higher dimensions)

In our applications, we will use the standard multivariate normal kernel.

$$K(z) = \frac{1}{\sqrt{(2\pi)^d}} \exp(z^T z)$$

where d is the dimensionality of the data. By using this kernel, the kernel density estimate takes the following form

$$f(x) \approx \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^d |H|}} \exp\left([x - D_i]^T H^{-1} [x - D_i]\right) \tag{6.2}$$

This estimate is non-parametric, but its smoothness hinges on the bandwidth parameter H. In the same way that width of bins in a histogram affect its smoothness, the kernel bandwidth H will affect the smoothness of the kernel density estimate.

When performing Bayesian diagnosis, the kernel density estimate is used to express the likelihood term

$$p(E|M) = f(E|M)$$
$$p(M|e) = \frac{p(E|M)p(M)}{\sum_M p(E|M)p(M)}$$

6.3.3 Bandwidth selection criterion

Selecting the bandwidth is not a trivial problem. The goal of bandwidth selection is to select a bandwidth that minimizes the asymptotic mean integrated square error (AMISE). One might recall the popular mean squared error (MSE) criterion

$$MSE = E\left[(\hat{f}(x) - f(x))^2\right]$$

where in this case $\hat{f}(x)$ is the kernel density estimate and $f(x)$ is the real density estimate. The MISE integrates this error over all values of x

$$MISE = \int E\left[(\hat{f}(x) - f(x))^2\right]\ dx$$

The MISE is generally intractable hence, the asymptotic approximation is used instead. In the univariate case, the AMISE is given by [73] as

$$AMISE = \frac{1}{4}h^4 R(f'')\mu_2(K)^2 + (nh)^{-1}R(K) \tag{6.3}$$

where n is the number of data points and h is the bandwidth (which, is a scalar for the univariate case), and where

$$R(K) \equiv \int K^2(z)\ dz$$
$$\mu_2(K) \equiv \int z^2 K(z)\ dz$$

[73] also defined a multivariate version of the AMISE criterion which is given as

$$\text{AMISE}\left[\hat{f}(\boldsymbol{x}; h)\right] = \frac{1}{4}\mu_2(K)^2\left[\text{vech}^T(H)\right]\boldsymbol{\Psi}\left[\text{vech}(H)\right] + \frac{R(K)}{n\sqrt{|H|}} \tag{6.4}$$

where

$$\mu_2(K) = \int_{\mathbb{R}^d} {z_i}^2 K(\boldsymbol{z})\ d\boldsymbol{z} \qquad \forall i$$
$$R(K) = \int_{\mathbb{R}^d} K(\boldsymbol{z})^2\ d\boldsymbol{z} < \infty$$
$$\boldsymbol{\Psi} = \int_{\mathbb{R}^d} \text{vech}(M)\text{vech}^T(M)\ \ d\boldsymbol{z}$$
$$M = 2\text{D}^2[f(\boldsymbol{z})] - \text{dg}\left(\text{D}^2[f(\boldsymbol{z})]\right)$$

In the multivariate AMISE criterion expression, $\text{vech}(H)$ takes the lower diagonal of H and strings it out column-wise into a vector. In addition, $\text{D}^2[f(\boldsymbol{z})]$ is the Hessian of $f(\boldsymbol{z})$, and the operator $\text{dg}(A)$ sets all off-diagonal elements of A to zero (the equivalent of the `diag(diag(A))` command in MATLAB).

6.3.4 Bandwidth selection techniques

The bandwidth selection criterion makes the assumption that we know the real density function $f(\boldsymbol{x})$. Obviously, if we knew the real density function $f(\boldsymbol{x})$, we would not need a kernel density estimate. The main idea of using the AMISE criterion however, is to select an optimal bandwidth for a distribution we know; this bandwidth will be close to optimal for similar distributions. In practice, the concern with selecting appropriate bandwidths has more to do with the amount of data and its general spread than it has to do with the specifics of the distribution.

Optimal bandwidths for multivariate normal distributions

The most popular method to select bandwidths is to use the optimal kernel density estimate for multivariate normal distributions. This bandwidth is defined as

$$H_N = \left(\frac{4}{n(d+2)} \right)^{\frac{2}{d+4}} \Sigma$$

where Σ is the covariance matrix of the target multivariate normal distribution. Now in practice Σ is not available to us, but the sample covariance matrix S can be easily obtained, resulting in the following bandwidth selection

$$H_N = \left(\frac{4}{n(d+2)} \right)^{\frac{2}{d+4}} S$$

If the relationships between the variables is linear, it is best to use the full covariance matrix estimate S. However, if the variables exhibit non-linear relationships, one may wish to set the off-diagonal elements of S to 0 so that the diagonal elements are all that remain.

Adaptive bandwidth estimation techniques

One problem with choosing a single bandwidth is that the distribution tends to be over-smoothed near the peaks of the distribution, but tends to be under-smoothed near the tails. This is analogous to the problem of histogram bins being well-estimated in regions of high probability, but very sparse and disjoint in regions of low probability.

In order to improve performance in these extreme cases, the bandwidth is modified so that the kernel function's height is proportional to the probability at that point. The reasoning behind this is that if the kernel's hight is proportional to the point's probability, then each kernel is expected to have the similar number of data points within its domain.

The bandwidth height is modified by setting a localized scalar λ_i in front of every individual bandwidth

$$H_i = \lambda_i^{-2/p} H_N \tag{6.5}$$

The scaling parameter λ_i can be calculated using a "pilot" density estimate $\hat{f}^p_{H_N}(x)$, which is obtained using the optimal normal bandwidth H_N.

$$\lambda_i = \left(\frac{\hat{f}^p_{H_N}(D[i])}{g} \right)^{\alpha}$$

where α is a user-defined parameter and g is the geometric mean of $\hat{f}^p_H(D[i])$, such that

$$\log(g) = \frac{1}{n} \sum_i \log \left[\hat{f}^p_H(D[i]) \right]$$

If one desires the kernel height to be proportional to the probability at that point, the user defined parameter α should be set to 1; however [74] found that $\alpha = 1$ was too aggressive

for univariate cases and suggested that for univariate cases one should set $\alpha = 0.5$ (which is a half-way point between the aggressively adaptive $\alpha = 1$ and the non-adaptive $\alpha = 0$). [63] also commented on this phenomenon after rigorous theoretical analysis, suggesting that the $\alpha = 0.5$ should also work well in higher dimensions. However, [63] also mentioned that adaptive kernel density estimation is useful for smaller sample sizes, but converges to the true density slower as the sample size increases. Thus, for larger sample sizes, smaller values of α should be used.

One can see that when setting $H_i = \lambda_i^{-2/p} H_N$, it results in a kernel density estimate given as

$$\begin{aligned}
\hat{f}(x) &= \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|H_i|^{1/2}} K\left(H_i^{1/2}(x - D_i)\right) \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|\lambda_i^{-2/p} H_N|^{1/2}} K\left(H_i^{1/2}(x - D_i)\right) \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{\lambda_i}{|H_N|^{1/2}} K\left(H_i^{1/2}(x - D_i)\right)
\end{aligned}$$

so that the height of each kernel is modified by λ_i.

6.4 Dimension Reduction

One point of difficulty for kernel density estimation is the problem of dimensionality. Much like its discrete counterpart, the difficulty of estimating kernel densities increases exponentially with respect to its dimension; this is referred to as the *curse of dimensionality*. The rate at which the difficulty increases is of the order $O\left[n^{-4/(d+4)}\right]$.

As an example, let us consider a one-dimensional system. In order to adequately estimate a kernel density, around 40 data points will be needed; the amount of required data points to achieve the same quality of estimation is shown in Table 6.2. As one might observe, for every additional dimension, the amount of data required is increased by a factor of $40^{1/5}$.

Table 6.2: Curse of dimensionality

Dimension	Required Data Points
1	40
2	84
3	175
4	366
5	765
6	1600

Because of this problem, kernel density applications for large systems must always consider a dimension reduction scheme. In this chapter, we will consider two main schemes,

independence assumptions and independent component analysis (ICA).

6.4.1 Independence assumptions

One method to reduce dimensionality is to introduce independence assumptions. For example, let us consider a six-dimensional system $E = [E_1, E_2, E_3, E_4, E_5, E_6]$. If, for mode M, the first three pieces of evidence can be considered independent of the second three, $[E_1, E_2, E_3] \perp [E_4, E_5, E_6]$ then we can calculate the joint probability as a product

$$p(E|M) = p(E_1, E_2, E_3|M)p(E_4, E_5, E_6|M) \qquad \text{if } [E_1, E_2, E_3]|M \perp [E_4, E_5, E_6]|M$$

Even though the joint probability is six-dimensional, we can break the problem down into two kernel density estimates: $p(E_1, E_2, E_3|M)$ and $p(E_4, E_5, E_6|M)$ which are both three-dimensional distributions. In this way, a six-dimensional problem was reduced to a three-dimensional one.

If one suspects some evidence to be independent, there is a test that can be used to verify whether this assumption can be made. The *mutual information criterion* (MIC) can be used to check for independence.

$$MIC(E_1, E_2) = \int_{E_1}\int_{E_2} p(E_1, E_2|M) \log\left(\frac{p(E_1, E_2|M)}{p(E_1|M)p(E_2|M)}\right) dE_1\ dE_2$$

The MIC can be calculated numerically by using kernel density estimates for $p(E_1, E_2|M)$, $p(E_1|M)$, and $p(E_2|M)$, and then numerically integrating the result over a suitable range of E_1, E_2.

It may not always be possible to break down the evidence into purely independent groups. However, a much more lenient conditional dependence assumption can also result in dimension reduction. Consider a case where all evidence $E = [E_1, E_2, E_3, E_4]$ is caused by a single underlying factor, which is best observed by E_1. If this is the case, then it is reasonable to break down the dimensions by conditioning with respect to E_1

$$\begin{aligned} p(E_1, E_2, E_3, E_4|M) &= p(E_1|M)p(E_2|E_1, M)p(E_3|E_1, M)p(E_4|E_1, M) \\ &= p(E_1|M)\frac{p(E_1, E_2|M)}{p(E_1|M)}\frac{p(E_1, E_3|M)}{p(E_1|M)}\frac{p(E_1, E_4|M)}{p(E_1|M)} \end{aligned}$$

By using conditional probability, highest dimension of this problem has been reduced from four to two. A variation of the MIC (the Conditional MIC or CMIC) can be used to test for conditional independence

$$\begin{aligned} CMIC(E_1, E_2) &= \int_{E_1}\int_{E_2} p(E_1, E_2|E_r, m) \log\left(\frac{p(E_1, E_2|E_r, M)}{p(E_1|E_r, M)p(E_2|E_r, M)}\right) dE_1\ dE_2\ dE_r \\ &= \int_{E_1}\int_{E_2} \frac{p(E_1, E_2, E_r|m)}{p(E_r|M)} \log\left(\frac{p(E_1, E_2, E_r|M)p(E_r|M)}{p(E_1|E_r, M)p(E_2|E_r, M)}\right) dE_1\ dE_2\ dE_r \end{aligned}$$

where E_r is the reference evidence.

6.4.2 Principal and independent component analysis

In addition to making independence assumptions, one can also attempt to explain the data with respect to a set of independent components. Independent component analysis (ICA) is a generalization of principal component analysis (PCA) and is a useful tool for dimension reduction. Both techniques assume a model where the data observations y are assumed to be linear combinations of latent variables f

$$y - \mu = At$$

The difference between PCA and ICA is that in PCA, the independent latent variables t are assumed to be Gaussian, and in ICA, they can follow any distribution (hence, the generalization). Both PCA and ICA aim to define the loading matrix A. The procedure for PCA is standard and will not be discussed. For ICA, a variety of algorithms exist, some of which have been discussed in [75]; the same author provided a MATLAB package in [76] for an algorithm that is both computationally efficient and reasonably accurate.

6.5 Missing Values

As in the discrete evidence case, it is possible for some values to be missing from the data. For the discrete case, Bayesian marginalization and the EM algorithm are typically the most popular solutions. For missing values in kernel density estimates, kernel density regression for the missing values is an effective and popular solution.

6.5.1 Kernel density regression

The zeroth-order (Nadaraya-Watson) method

Kernel density regression was first proposed by [77], shortly after the widely cited paper on kernel density estimation [61]. It can be derived by first noting the following regression function

$$\hat{y} = g(x) = \frac{\int y f(y, x) dy}{f(x)}$$

We let the joint probability estimate for $f(y, x)$ be

$$\hat{f}(y, x) = \frac{1}{n|H_x|^{1/2}|H_y|^{1/2}} \sum_{i=1}^{n} K\left(H_x^{-1/2}[X_i - x]\right) K\left(H_y^{-1/2}[Y_i - y]\right)$$

$$\hat{f}(x) = \frac{1}{n|H_x|^{1/2}} \sum_{i=1}^{n} K\left(H_x^{-1/2}[X_i - x]\right)$$

where $K(x)$ is a kernel function (such as the standard multivariate normal distribution). The kernel functions are set to be independent between x and y to facilitate the integration

over y (independent kernels do not suggest that x and y are independent, but that the bandwidth matrix H is simply diagonal).

$$\begin{aligned}\int y\hat{f}(y,x)dy &= \int y\frac{1}{n|H_x|^{1/2}|H_y|^{1/2}}\sum_{i=1}^{n}K\left(H_x^{-1/2}[X_i-x]\right)K\left(H_y^{-1/2}[Y_i-y]\right)dy \\ &= \frac{1}{n|H_x|^{1/2}}\sum_{i=1}^{n}K\left(H_x^{-1/2}[X_i-x]\right)\int\frac{y}{|H_y|^{1/2}}K\left(H_y^{-1/2}[Y_i-y]\right)dy \\ &= \frac{1}{n|H_x|^{1/2}}\sum_{i=1}^{n}K\left(H_x^{-1/2}[X_i-x]\right)Y_i\end{aligned}$$

This results in the following estimator for y

$$\begin{aligned}g(x) &= \frac{\frac{1}{n|H_x|^{1/2}}\sum_{i=1}^{n}K\left(H_x^{-1/2}[X_i-x]\right)Y_i}{\frac{1}{n|H_x|^{1/2}}\sum_{i=1}^{n}K\left(H_x^{-1/2}[X_i-x]\right)} \\ &= \frac{\sum_{i=1}^{n}K\left(H_x^{-1/2}[X_i-x]\right)Y_i}{\sum_{i=1}^{n}K\left(H_x^{-1/2}[X_i-x]\right)}\end{aligned} \tag{6.6}$$

This result amounts to a weighted average of historical Y values based on the proximity of the corresponding X values to the query value x; hence, it is a *locally weighted average*.

The first-order method

The Nadaraya-Watson method is a fairly popular method of non-parametric (or kernel density) regression. However, it has been shown to be biased toward flat functions of y with respect to x. In order to reduce this bias, the first-order method was proposed. Here, instead of using a locally-weighted average, we wish to use a locally-weighted linear model

$$y_i = \alpha + \beta^T(X_i - x) + \varepsilon_i$$

We can arrive to the ordinary least squares solution by setting

$$Z_i = \begin{bmatrix} 1 \\ X_i - x \end{bmatrix}$$

and then performing the following operation to obtain a linear model

$$\begin{bmatrix} \hat{\alpha}(x) \\ \hat{\beta}(x) \end{bmatrix} = \left[\sum_{i=1}^{n} Z_i Z_i^T\right]^{-1}\left[\sum_{i=1}^{n} Z_i Y_i\right]$$

Due to the way this problem was posed (being centered around x when we defined Z), $\hat{\alpha}(x)$ serves as our estimate of y.

$$\hat{y} = \hat{g}(x) = \hat{\alpha}(x)$$

However, for the kernel density variation, we use the kernel as a weighting function to create a locally weighted linear solution

$$\begin{bmatrix} \hat{g}(x) \\ \hat{\beta}(x) \end{bmatrix} = \left[\sum_{i=1}^{n} K\left(H_x^{-1/2}[X_i - x] \right) Z_i Z_i^T \right]^{-1} \left[\sum_{i=1}^{n} K\left(H_x^{-1/2}[X_i - x] \right) Z_i Y_i \right] \tag{6.7}$$

The locally weighted linear solution does not suffer as much from the bias toward flat regression estimates. Higher-order solutions also exist, but their improvement over the first-order method tends to be quite minimal.

6.5.2 Applying kernel density regression for a solution

Let us consider a data set X wherein some data entries are incomplete

$$X = \begin{bmatrix} X_c \\ X_{ic} \end{bmatrix}$$

Within each incomplete data entry, there are values that are present, denoted as z, and values that are missing denoted as y. For each incomplete data entry, $X_{ic}[i]$, we use kernel density regression $\hat{g}(z|X_c)$ on z (based on the complete data set X_c) to estimate the missing values y.

$$\begin{aligned} \hat{y} &= \hat{g}(z|X_c) \\ \hat{X}_{ic}[i] &= [z, \hat{y}] \end{aligned}$$

We now have a complete data estimate

$$\hat{X}_c = \begin{bmatrix} X_c \\ \hat{X}_{ic} \end{bmatrix}$$

This complete data estimate can either be used as the kernel density estimate, or we could perform another iteration of the estimation procedure.

$$\begin{aligned} \hat{y} &= \hat{g}\left(z | \left\{ \hat{X}_c \setminus \hat{X}_{ic}[i] \right\} \right) \\ \hat{X}_{ic}[i] &= [z, \hat{y}] \end{aligned}$$

where $\left\{ \hat{X}_c) \setminus \hat{X}_{ic}[i] \right\}$ denotes as set difference (or simply that we remove $\hat{X}_{ic}[i]$ from the dataset $\hat{X}_c$. This iterative scheme converges fairly quickly; it seldom requires more than ten iterations. For most intents and purposes, a single iteration will yield an adequate result.

6.6 Dynamic Evidence

Previously, it was shown that the likelihood for autodependent discrete evidence can be obtained as

$$p(E^t | E^{t-1}, m_k) = \frac{n(E^t, E^{t-1}, m_k)}{n(E^{t-1}, m_k)} \tag{6.8}$$

where $n(E^t, E^{t-1}, m_k)$ is the number of times that E^t, E^{t-1} and m_k are jointly observed in history while $n(E^{t-1}, m_k)$ is the number of times that $n(E^{t-1})$ and m_k) are jointly observed in the history. This solution was modified in the previous chapters to include prior samples in order to prevent possible division by zero.

A very similar solution can be made by using the kernel density method to estimate the likelihood. In order to condition on both E^t and E^{t-1} the rule of conditioning is applied

$$p(Y|X) = \frac{p(X \cap Y)}{p(X)}$$

By applying the rule of conditioning to the kernel density estimation solution is given as

$$p(E^t|E^{t-1}, M^t) = \frac{p(E^t, E^{t-1}|M^t)}{p(E^{t-1}|M^t)} \tag{6.9}$$

This result requires two kernel density estimates to be evaluated:

1. $p(E^t, E^{t-1}|M^t)$ for the likelihood of joint present and past evidence (given the mode M^t)
2. $p(E^{t-1}|M)$ for the likelihood of past evidence (given the mode M^t)

The resulting ratio of likelihoods can be used in the same manner as a likelihood; for example, when used in Bayesian diagnosis, the dynamic solution is applied as follows:

$$p(E^t|E^{t-1}, M^t) = \frac{p(E^t, E^{t-1}|M^t)}{p(E^{t-1}|M^t)}$$
$$p(M^t|E^t, E^{t-1}) = \frac{p(E^t|E^{t-1}, M^t)p(M^t)}{\sum_M p(E^t|E^{t-1}, M^t)p(M^t)}$$

Dimensionality reduction

The use of dynamic evidence however, has the problem of adding dimensionality to the data. Thus, when applying a dynamic evidence solution, dimensionality reduction techniques (such as ICA and dependence analysis) are even more vital to perform. Because of the curse of dimensionality, it is desirable to test whether it is necessary to include past evidence in the likelihood. Again, this test can be performed using the MIC, but with a focus on past and present evidence

$$MIC(E_k^t, E_k^{t-1}) = \int_{E_k^t} \int_{E_k^{t-1}} p(E_k^t, E_k^{t-1}|m) \log \left(\frac{p(E_k^t, E_k^{t-1}|M)}{p(E_k^t|M)p(E_k^{t-1}|M)} \right) dE_k^t \; dE_k^{t-1}$$

If $MIC(E_k^t, E_k^{t-1})$ is a small number (generally less than 0.2), then we do not need to include past evidence for this particular evidence source E_k.

Chapter 7

Accounting for Sparse Modes Within the Data

7.1 Introduction

In this chapter, we consider how to address the problem of diagnosing a system when modes are missing entirely. Recall that operating modes must include information of all system components; however, as the number of components increases, the number of possible operating modes will tend to grow exponentially. Because of this, modes that are missing entirely is a very pertinent issue to systems having more than a small number of components (for example, an eight component system will have at least $2^8 = 256$ modes).

This chapter discusses two approaches one can take to deal with the problem of missing operating modes.

1. The first approach is to focus on diagnosing the state of each component; this approach is fairly easy to perform in practice, as it is a simple re-structuring of the original diagnosis problem.

2. The second approach is to use process modelling and bootstrapping to simulate the missing faulty scenarios; this approach is difficult to implement in practise as it requires sufficient process knowledge to simulate faulty behaviour. Furthermore, if the system is large, a very large number of simulations will be required in order to cover every possible mode.

The two techniques are independent and do not interfere with each other, consequently, applying both techniques simultaneously requires no modification of either technique.

7.2 Algorithms

This chapter discusses two separate algorithms: the first algorithm focuses on diagnosis in component space the second algorithm focuses on generating data for unencountered modes.

7.2.1 Algorithm for component diagnosis

Overview

Fault diagnosis focuses on collecting data to estimate the distributions $p(E|M)$; however, if there is a large number of possible modes in M, this will result in the historical data having to be divided into a large number of modes. Some modes will be well represented in their data, but many of them will have little data or no data at all. By contrast, if we evaluate $p(E|C)$, where C represents the state of the component of interest, there should be data from each state of the component.

Consider a basic control loop in Figure 7.1. Here, the components of interest are the control valve, the actuator, the sensor and the process (which could have multiple components itself).

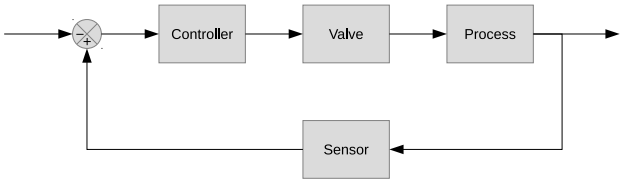


Figure 7.1: Overall Algorithm

The overall aim of component diagnosis is to diagnose the state of each component, and then diagnose the mode. The probability of the state for a component i can be calculated according to Bayes' rule.

$$p(C^i) = \frac{p(E|C^i)p(C^i)}{\sum_{C^i} p(E|C^i)p(C^i)}$$

If this system had three states for each component, the total number of modes (and required distributions) would be $3^4 = 81$. By contrast, considering each component one at a time would require the estimation of $4 \times 3 = 12$ distributions, which is a dramatic reduction in terms of the diagnosis space. While it may be a challenge to find data that would correspond to 81 modes (and hence 81 different conditions), it is much easier to find data from which each component state is realized so that only 12 conditions need to be realized. This result leads us into the main advantages of the component diagnosis technique:

1. **Reduction in problem complexity for modes:** The primary reason we consider the component diagnosis technique is that it reduces the number of modes we have to diagnose. Using the mode-based diagnosis approach, the complexity of the problem grows exponentially with each new component. By contrast, using the component-based diagnosis approach, the complexity of the problem grows linearly with each new component. An eight component system with two modes each will have $2^8 = 256$ modes to diagnose under the mode-based approach, but the same system will have $2 \times$

$8 = 16$ modes to diagnose under the component-based approach. Since the component-based approach has one sixteenth as many modes as the mode-based approach, there is a much smaller probability that modes will be missing if the component-based approach is applied.

2. **Reduction in problem complexity for evidence:** Each piece of evidence tends to be sensitive toward a few components. Because the mode-based approach considers all components, all available evidence should be used; this can lead to fairly high-dimensional distributions. However, if we adopt the component-based approach, we only need to consider the evidence that is sensitive toward that component; the rest can be discarded. This allows us to effectively reduce the dimensionality of the evidence; recall that in Chapter 6, evidence dimension space was a problem for kernel density estimation, and even more for discrete methods. Using the component-based approach allows us to reduce evidence dimensionality in a manner that is easier than testing and assuming independence (which was the solution suggested in Chapter 6).

While the component based approach has its merits, it also has one key drawback. The component based approach assumes that the component states are independent of each other, which is not a true assumption if for example, a problem in once component tends to cause problems in another component. In such cases the mode based approach will outperform the component based approach if sufficient data is available for each mode.

Selecting monitors for components

When diagnosing between modes, any available sensor or monitor could be helpful as long as it can help distinguish at least one mode from another. Similarly, when diagnosing between components, one chooses a set of sensors or monitors that are sensitive to changes between any states in that component. Because modes include the states of all components, the number of sensors needed to distinguish a single component will tend to be fewer than the monitors needed to distinguish between modes.

In component diagnosis, it is important only to select monitors and sensors that are sensitive to changes in that particular component. Other sensors will be sensitive to other components, and including them may result in misleading information especially if not every mode is realized in the data. The use of fewer sensors and monitors for each component has the added advantage of dimension reduction. In most cases, using the MIC criterion to reduce dimensionality is unnecessary when component based diagnosis is used. One can use the *false negative criterion* to determine how sensitive a particular sensor or monitor is to a mode, which can be calculated by following a series of steps

1. Select a component of interest. For example, let us consider a four-component system $[C^1, C^2, C^3, C^4]$, where the first component C^1 is of interest.

2. Search for historical data for situations where only the component of interest changes. One would select modes that have different values in C^1 but the values of C^2, C^3, C^4 remain constant. It is best to select a set of constant values C^2, C^3, C^4 where abundant data is present for all values of C^1. Note that because we are only testing one evidence source at a time, the dimension is small making data requirements easier to meet (100 data points is a good sample size, but 40 will suffice). This means we can be quite flexible with data collection for this purpose.

3. Group the data according to the different states in C^1; data from each state value in C^1 will be used to evaluate the likelihood $p(E^i|C, D)$ (where D is the historical data).

4. Select a monitor/sensor of interest E^i and obtain corresponding data D^i for that instrument. Evaluate the set of historical data likelihoods $p(D^i|c_k^1, D)$ based on the monitor/sensor of interest.

5. Use the likelihoods from one state $p(D^i|c_k^1)$ to diagnose the state based on the maximum likelihood ($\hat{C}^1$ is the diagnosed state with the maximum likelihood).

6. Determine how frequently the data from state c_k^1 is diagnosed as some other state $n[\hat{C}^1 \neq c_k^1|c_k^1]_{E^i}$ when E^i is used to make the decision. This is referred to as a false negative frequency.

7. Obtain the false negative probability for each component state k by normalizing the false negative frequency

$$P[\hat{C}^1 \neq c_k^1|c_k^1]_{E^i} = \frac{n[\hat{C}^1 \neq c_k^1|c_k^1]}{n[c_k^1]_{E^i}}$$

8. The false negative rate is obtained as

$$FN_{E^i} = \frac{1}{n-1}\sum_{k=1}^{n} P[\hat{C}^1 \neq c_k^1|c_k^1]_{E^i} \tag{7.1}$$

   where n is the number of states component C^i can take. $FN_{E^i} = 1$ indicates that E^i is perfectly uninformative, and where $FN_{E^i} = 0$ indicates that E^i is a perfect classifier.

9. Based on FN_{E^i} decide whether or not E^i should be included to estimate the state of the component of interest C^1. Generally, it is best to select E^i when FN_{E^i} is low (for example, less than 0.5).

10. Repeat steps 4-9 for other monitors/sensors

11. Repeat steps 1-10 for other components of interest

Constructing and evaluating probabilities

After the included evidence has been selected for each component, we can construct their respective likelihood functions (either by discrete means or kernel density estimation). The simplest manner to construct the likelihood function $p(E|C)$ is to estimate the function based on data from all modes where C has the desired state. However, modes that occur frequently will be given a lot of weight in this function. It is often better to select the most likely mode where the component state $C(k)$ is true (denoted as $M \supset C$) so that

$$p(E|C) = \max_{M \supset C} p(E|M)$$

Note that if all of the evidence E is selected for every component, then the component diagnosis results will be the same as the mode diagnosis results if the component states are independent. The improvement of the component diagnosis result stems from the fact that only evidence sensitive to the component is used to diagnose the component's state. When this is done, it is possible to evaluate any of the possible modes so long as all states for each component are present in the data.

The posterior probability of the component state is obtained by using Bayes' Theorem

$$p(C|E) = \frac{p(E|C)p(C)}{\sum_k p(E|c_k)p(c_k)} \tag{7.2}$$

where

$$p(C) = \sum_{M \supset C} p(M)$$

One can diagnose the most probable state as the true component. Once all the components $C^1, C^2, \ldots C^p$ are diagnosed (as $\hat{C}^1, \hat{C}^2, \ldots, \hat{C}^p$) we can diagnose the mode that contains the appropriate component states.

$$\hat{m} = [\hat{C}^1, \hat{C}^2, \ldots, \hat{C}^p]$$

By assuming all component states are independent, it is also possible to evaluate the posterior probability of the modes

$$p(M|E) = \prod_k p(C^k \subset M|E) \tag{7.3}$$

where $C^k \subset m$ indicates that C^k is contained in the mode M, or equivalently, C^k takes the value specified by that component of M. Obviously, the most probable mode is the one that contains all diagnosed component states.

7.2.2 Algorithm for bootstrapping new modes

Bootstrapping for new modes is done in a manner similar to the technique presented in Qi and Huang [48]; however, in this chapter, the underlying fault parameters are varied in order to simulate new scenarios. Considering a control loop with various components, the algorithm can be done by taking the following steps which are briefly described below:

1. Create a model structure for each system component which includes all relevant fault parameters and unknown parameters if any.
2. If there are unknown parameters obtain data for model identification (for the most reliable results, open-loop testing) and use gray-box modelling to identify unknown parameters.
3. Obtain residual error information by subtracting predicted output from observed output.
4. Whiten residual errors by identifying an AR model and applying its inverse to the residual errors.
5. Estimate a kernel density function from the whitened residual errors.
6. Simulate new process data using the identified model by manipulating the fault parameters. Disturbances can be generated by sampling from the kernel density function (smoothed bootstrapping) and applying the AR model to the sampled estimates.
7. Apply monitoring algorithms to both simulated and real process data. Monitor results are used as reference data for Bayesian diagnosis.

Step 1: Create model structures

This step is highly process dependent. If there are multiple components to the system (such as the case with a control loop), each of the components has to be modelled. The model structure should be given such that fault-related parameters can be easily seen and manipulated. An example of this is given later on with respect to the hybrid tank system. Because the model in question must make use of parameters that have physical meaning, the model structure must be derived using a *white box* or *gray box* approach, where *white box* models are based on first principles and have no unknown parameters, and where *gray box* models are constructed based on first principles, but can have simplifications and unknown parameters (to be estimated using data).

Step 2: Use data to identify gray box models

If one cannot construct a white box model (which is usually the case), one can use data to estimate unknown parameters in a gray box model. Consider a dynamic model $f(x, u, \Theta)$ with states x, inputs u and unknown parameters Θ.

$$\frac{dx}{dt} = f(x, u, \Theta) + \varepsilon_x$$

This model is subject to an observation function $h(x)$ to yield the observed values y

$$y = h(x) + \varepsilon_y$$

where the observation function $h(x)$ is assumed to be known. The predicted values of y can be obtained by using an ordinary differential equation solver (such as the RK45 method) to solve for x given u, and then predict y using the observation function $h(x)$. The estimated value of Θ can then be obtained by minimizing an error expression

$$\begin{aligned}\hat{x}(t) &= \text{RK45}\left[f(x(t-1), u(t-1), \Theta)\right] \\ \hat{\Theta} &= \arg\min_{\Theta} \sum_{t} \left[h(\hat{x}(t)) - y(t)\right]^T R^{-1} \left[h(\hat{x}(t)) - y(t)\right]\end{aligned}$$

where R is a positive-definite (often diagonal) weighing matrix that represents the noise variance of each instrument. The purpose of R is to give an appropriate weight to each element of the observation vector $y(t)$. If $y(t)$ has only one element, $R = 1$ is sufficient, and if all elements of $y(t)$ take the same units, $R = I$ will also be sufficient. Otherwise, one might want to take the variance of each element in y in order to obtain a suitable metric for scaling.

Step 3: Obtain residual errors

Residual error calculations require estimates of the hidden state, which can be obtained using Kalman filtering. In general, the state models are nonlinear, thus an ordinary Kalman filter will not suffice. It is recommended that more advanced techniques such as the Extended Kalman filter (EKF), the Unscented Kalman filter (UKF), the Ensemble Kalman Filter (EnKF), or the Particle Filter (PF) are used. Due to a combination of easy computation and effectiveness, the Unscented Kalman Filter (UKF) a popular choice for nonlinear state estimation problems.

The UKF estimates the state given a system that has the following conditions:

$$\begin{aligned}\frac{dx}{dt} &= f(x, u, \Theta) + \varepsilon_x \\ y &= h(x) + \varepsilon_y \\ \varepsilon_x &\sim N(0, Q) \\ \varepsilon_y &\sim N(0, R)\end{aligned}$$

where $\varepsilon_x \sim N(0, Q)$ indicates that ε_x is Gaussian white noise with mean zero and covariance Q (similar conditions are assumed for $\varepsilon_y \sim N(0, Q)$). If one already has suitable values of Q and R, the residual errors of x and y can be obtained via the unscented Kalman filter. The residual errors on x (denoted as $\hat{\varepsilon}_x$) can be obtained using

$$\begin{aligned}x(t) &= \text{UKF}\left[f(x(t-1), u(t-1)), p(t-1), h(x), y(t), Q, R\right] \\ \hat{x} &= \text{RK45}\left[f(x(t-1), u(t-1), \Theta)\right] \\ \hat{\varepsilon}_x &= x - \hat{x}\end{aligned}$$

where x is the more reliable estimate (as it uses observations y), and $\hat{x}$ is the predicted value without considering noise (a less reliable estimate which does not use y).

Meanwhile the residual errors on y (denoted as $\hat{\varepsilon}_y$) can be obtained using

$$\hat{y} = h(x(t))$$
$$\hat{\varepsilon}_y = y - \hat{y}$$

The values for Q and R are often used as tuning parameters which express the reliability of the model and observations respectively. Large values in Q assume the model is less reliable, but large values in R assume that the observations are unreliable. It is also possible to estimate Q and R from data using a technique that is similar to the EM algorithm. The technique makes use of the following steps:

1. Start with initial values for Q and R. If the system changes slowly and the model is fairly accurate, a good initial estimate of R can be obtained using

   $$R_0 = \frac{1}{2(n-1)}\text{diag}\left[\sum_{i=2}^{n}(y(i) - y(i-1))^2\right]$$

   where $\text{diag}(x)$ is the same as the MATLAB command `diag(x)` which takes a vector x and constructs a diagonal matrix out of it. An initial value of Q can be obtained by analysing the R_0 and the observation function h, generally, it is best to choose large values for Q initially, so that the observations carry more weight than the model.

2. Use Kalman filtering to estimate states given values chosen for Q and R

   $$f(x) = f(x, u(t-1), \Theta)$$
   $$x(t) = \text{UKF}(f(x), x(t-1), p(t-1), h(x), y(t), Q_0, R_0)$$

   where $u(t-1)$ is assumed to be a constant over the given time interval.

3. Estimate a new value for Q using

   $$\hat{x} = \text{RK45}\left[f(x(t-1), u(t-1), \Theta)\right]$$
   $$Q_1 = \frac{1}{n}\sum_{i=1}^{n}[x(i) - \hat{x}(i)] \times [x(i) - \hat{x}(i)]^T$$

4. Estimate a new value for R using

   $$\hat{y} = h(\hat{x})$$
   $$R_1 = \frac{1}{n}\text{diag}\left[\sum_{i=1}^{n}(y(i) - \hat{y}(i))^2\right]$$

5. Repeat steps 2-4 until the log likelihood of the data L converges

$$
\begin{aligned}
L &= L_x + L_y \\
L_x &= -\frac{n}{2}\log\left[(2\pi)^{d_x}|Q|\right] - \frac{n}{2}\sum_i [x(i)-\hat{x}(i)]^T Q^{-1}[x(i)-\hat{x}(i)] \\
L_y &= -\frac{n}{2}\log\left[(2\pi)^{d_y}|R|\right] - \frac{n}{2}\sum_i [y(i)-\hat{y}(i)]^T R^{-1}[y(i)-\hat{y}(i)]
\end{aligned}
$$

where d_x is the dimension of x, and d_y is the dimension of y.

After obtaining Q and R, one can further tune these values to suit their needs. Scaling for smaller values of Q will result in smoother state estimates but will be less responsive to observations; conversely scaling for larger values of Q result in rougher state estimates but will be more responsive to observations.

Step 4: Whitening residual errors

Bootstrapping (and its smoothed counterpart) requires that the residual errors be independent and identically distributed (IID). Often times, the residual errors are autocorrelated but can be whitened by applying an auto-regressive (AR) model. This is done by first identifying and AR model for both $\hat{\varepsilon}_x$ and $\hat{\varepsilon}_y$

$$
\begin{aligned}
\hat{\varepsilon}_x(t) + A_1\hat{\varepsilon}_x(t-1) + \ldots + A_n\hat{\varepsilon}_x(t-n) &= \hat{\varepsilon}_x^w(t) \\
\hat{\varepsilon}_x &= \frac{1}{A(z)}\hat{\varepsilon}_x^w
\end{aligned}
$$

The whitened residuals are obtained by inverting the model so that

$$\hat{\varepsilon}_x^w = A(z)\hat{\varepsilon}_x$$

Note that the same whitening procedure should be performed on $\hat{\varepsilon}_y$.

Step 5: Kernel density estimation

Recall that the kernel density estimate can be calculated as

$$f(x) \approx \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^d|H_i|}} \exp\left([x-D_i]^T H_i^{-1}[x-D_i]\right)$$

where D represents the data points (in this case $D = \hat{\varepsilon}_x^w$, d is the dimension of x, and H_i is the bandwidth at x_i. Also recall that if one is using a uniform bandwidth, $H_i = H$ can be calculated using the normal bandwidth reference rule

$$H_N = \left(\frac{4}{n(d+2)}\right)^{\frac{2}{d+4}} S \tag{7.4}$$

where S is the sample covariance matrix of the data $D = \hat{\varepsilon}_x^w$ or $D = \hat{\varepsilon}_y^w$. The kernel density estimate consists of data and bandwidth matrices corresponding to the data points. Note that the adaptive bandwidth technique mentioned in Chapter 6 can also be used.

For this step, since the data $D = \hat{\varepsilon}_x^w$ and $D = \hat{\varepsilon}_y^w$ are already obtained, one simply needs the bandwidth matrix to complete the kernel density estimate.

Step 6: Simulate new data via smoothed bootstrapping

The kernel density estimate forms the basis of the smoothed bootstrap. One can sample from the kernel density estimate by means of a two-step process

1. Randomly select a data point $\hat{\varepsilon}_x^w(i)$ or $\hat{\varepsilon}_y^w(i)$ from the history, where i is a random integer between 1 and n with uniform probability.

2. Add Gaussian noise to $\hat{\varepsilon}_x^w(i)$ or $\hat{\varepsilon}_y^w(i)$ with mean of zero and covariance H_i. Note that adding Gaussian noise samples a Gaussian distribution centred around the selected data point instead of sampling the data point itself; by sampling the kernel function around the data point, ordinary bootstrapping is converted to *smoothed bootstrapping*.

The sampling and noise-adding is repeated for the number of times the simulation is desired. After smoothed bootstrapping, the AR filter is used to generate disturbances that act similarly to the original case.

$$\hat{\varepsilon}_x = \frac{1}{A(z)}\hat{\varepsilon}_x^w$$
$$\hat{\varepsilon}_y = \frac{1}{A(z)}\hat{\varepsilon}_y^w$$

The disturbance sequences can be applied to the process and observation models ($f(x, u, \Theta)$ and $h(x)$ respectively) to generate new samples for y. New modes can be created by varying the parameters Θ in the gray box model $f(x, u, \Theta)$ that corresponds to different faults, but the step of generating noise from bootstrapping remains the same for each new simulated mode.

Step 7: Apply monitoring algorithms

The monitoring algorithms are process specific, and can be applied to the additional simulated data in the exact same manner as the original data.

7.3 Illustration

In this illustration, we will consider the hybrid tank system which will also be presented in the practical application later on (in Section 7.4); a schematic is available in Figure 7.2. The hybrid tank system has four components each with two states, resulting in 16 modes in all. The four components are the flow sensor into tank 1 (FM_1), the flow sensor into

tank 2 (FM_2), the valve between tanks 1 and 2 (V_1), and the valve between tanks 2 and 3 (V_2). Modes will be described in terms of bias in FM_1, FM_2 (B_1, B_2 respectively) and leaks caused by opening the valves V_1, V_2 (L_1, L_2 respectively); the mode vector in this example is $[B_1, B_2, L_1, L_2]$ where each component can take the state 0 (where the problem does not exist) or 1 (where the problem does exist).

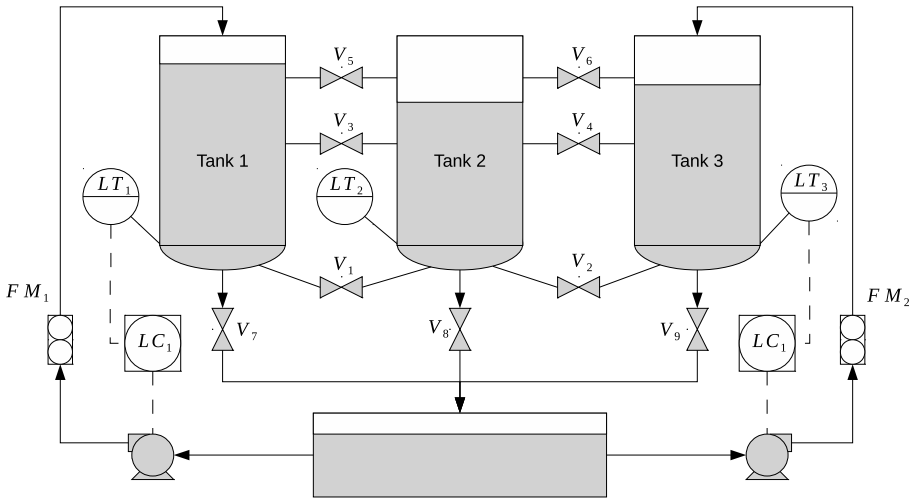


Figure 7.2: Hybrid tank system

The monitor selected for this system consists of an augmented Kalman filter and calculated pump model prediction errors. For the original filter, the state model is given as follows

$$\begin{aligned}
\frac{d\,X_1}{d\,t} &= A_c^{-1}\left[\frac{U_1}{B_1} - C_1\left[X_1^{1/2} + h_r\right] + L_1\frac{X_2 - X_1}{|X_2 - X_1|^{1/2}}\right] + \varepsilon_{X_1} \\
\frac{d\,X_2}{d\,t} &= A_c^{-1}\left[-C_2\left[X_2^{1/2} + h_r\right] + L_1\frac{X_2 - X_1}{|X_2 - X_1|^{1/2}} + L_2\frac{X_2 - X_3}{|X_2 - X_3|^{1/2}}\right] + \varepsilon_{X_2} \\
\frac{d\,X_3}{d\,t} &= A_c^{-1}\left[\frac{U_2}{B_2} - C_3\left[X_3^{1/2} + h_r\right] + L_1\frac{X_2 - X_3}{|X_2 - X_3|^{1/2}}\right] + \varepsilon_{X_3} \\
Y &= IX + \varepsilon_Y
\end{aligned}$$

where the state vector X consists of the three level indicators, and the input vector U consists of the flow rate measurements into Tanks 1 and 3. The state is augmented to include four additional states, B_1, B_2, L_1, L_2 representing bias and leak parameters which

are ideally constant

$$\frac{d\ B_1}{d\ t} = 0$$
$$\frac{d\ B_2}{d\ t} = 0$$
$$\frac{d\ L_1}{d\ t} = 0$$
$$\frac{d\ L_2}{d\ t} = 0$$

When augmented, the additional state covariance entries in Q are set to be small values as the monitored values are believed to be relatively constant. Small values in Q also mean that the monitored results are smooth.

The four monitored values are obtained by applying the unscented Kalman filter to the augmented model. A MATLAB implementation of the unscented Kalman filter is available on the MATLAB file exchange, courtesy of Yi Cao of Cranfield University, and is presented below. For the original version, one can visit
http://www.mathworks.com/matlabcentral/fileexchange/18217-learning-the-unscented-kalman-filter

```
function [x,P]=ukf(fstate,x,P,hmeas,z,Q,R)
% UKF   Unscented Kalman Filter for nonlinear dynamic systems
% [x, P] = ukf(f,x,P,h,z,Q,R) returns state estimate, x and state covariance, P
% for nonlinear dynamic system (for simplicity, noises are assumed as additive):
%           x_k+1 = f(x_k) + w_k
%           z_k   = h(x_k) + v_k
% where w ¬ N(0,Q) meaning w is gaussian noise with covariance Q
%       v ¬ N(0,R) meaning v is gaussian noise with covariance R
% Inputs:   f: function handle for f(x)
%           x: "a priori" state estimate
%           P: "a priori" estimated state covariance
%           h: function handle for h(x)
%           z: current measurement
%           Q: process noise covariance
%           R: measurement noise covariance
% Output:   x: "a posteriori" state estimate
%           P: "a posteriori" state covariance
%
%
% By Yi Cao at Cranfield University, 04/01/2008
%
L=numel(x);                                 %numer of states
m=numel(z);                                 %numer of measurements
alpha=1e-3;                                 %default, tunable
ki=0;                                       %default, tunable
beta=2;                                     %default, tunable
lambda=alpha^2*(L+ki)-L;                    %scaling factor
c=L+lambda;                                 %scaling factor
Wm=[lambda/c 0.5/c+zeros(1,2*L)];           %weights for means
Wc=Wm;
Wc(1)=Wc(1)+(1-alpha^2+beta);               %weights for covariance
c=sqrt(c);
X=sigmas(x,P,c);                            %sigma points around x
[x1,X1,P1,X2]=ut(fstate,X,Wm,Wc,L,Q);          %unscented transformation of process
% X1=sigmas(x1,P1,c);                         %sigma points around x1
```

```
% X2=X1-x1(:,ones(1,size(X1,2)));            %deviation of X1
[z1,Z1,P2,Z2]=ut(hmeas,X1,Wm,Wc,m,R);       %unscented transformation of measurments
P12=X2*diag(Wc)*Z2';                        %transformed cross-covariance
K=P12/P2;
x=x1+K*(z-z1);                              %state update
P=P1-K*P12';                                %covariance update

function [y,Y,P,Y1]=ut(f,X,Wm,Wc,n,R)
%Unscented Transformation
%Input:
%         f: nonlinear map
%         X: sigma points
%        Wm: weights for mean
%        Wc: weights for covraiance
%         n: numer of outputs of f
%         R: additive covariance
%Output:
%         y: transformed mean
%         Y: transformed smapling points
%         P: transformed covariance
%        Y1: transformed deviations

L=size(X,2);
y=zeros(n,1);
Y=zeros(n,L);
for k=1:L
    Y(:,k)=f(X(:,k));
    y=y+Wm(k)*Y(:,k);
end
Y1=Y-y(:,ones(1,L));
P=Y1*diag(Wc)*Y1'+R;

function X=sigmas(x,P,c)
%Sigma points around reference point
%Inputs:
%       x: reference point
%       P: covariance
%       c: coefficient
%Output:
%       X: Sigma points

A = c*chol(P)';
Y = x(:,ones(1,numel(x)));
X = [x Y+A Y-A];
```

In addition, two more monitors are included to estimate flow meters using pump speed signals. Under normal conditions, the output error (OE) model is estimated to predict flow rates

$$U(t) = \frac{B(z)}{C(z)} S(t)$$

where $S(t)$ is the pump speed. The two additional monitors estimate bias as

$$B_1(t) = \frac{U_1(t)}{\frac{B(z)}{C(z)} S(t)}$$

$$B_2(t) = \frac{U_2(t)}{\frac{B(z)}{C(z)} S(t)}$$

7.3.1 Component-based diagnosis

Offline Step 1: Select components and choose reference modes

In the case of our data, the most frequent mode is the no fault mode $[0,0,0,0]$, thus for the first component, we will be evaluating the false negative criterion (FN) between $m_1 = [0,0,0,0]$ and $m_9 = [1,0,0,0]$. Likewise, for the second component, we will be evaluating FN between $m_1 = [0,0,0,0]$ and $m_5 = [0,1,0,0]$; this procedure is then repeated for the third and fourth components which compare $m_3 = [0,0,1,0]$ and $m_2 = [0,0,0,1]$ respectively to $m_1 = [0,0,0,0]$. The list of modes for this system is given as follows:

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \\ m_7 \\ m_8 \\ m_9 \\ m_{10} \\ m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{15} \\ m_{16} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \texttt{ModeBV} \tag{7.5}$$

Let us consider the first component, where the reference modes are $m_1 = [0,0,0,0]$ and $m_9 = [1,0,0,0]$. We assume that MATLAB is used, and that data is already sectioned into different modes in a cell array labelled `Data` so that `Data{1}` contains data for $m_1 = [0,0,0,0]$ and `Data{9}` contains data for $m_9 = [1,0,0,0]$.

Offline Step 2: Calculate FN criterion for each instrument

The FN criterion for each instrument can be calculated according to the MATLAB code below which follows steps 1-10 under *Selecting monitors for components* in Section 7.2.1.

```
function FN = FalseNegativeCriterion(Data)

ne = length(Data{1}(1,:)); %find the dimension of the data
ns = length(Data); %find the number of states for this component

%For each piece of evidence
for e = 1:ne
    %For each state
    for i = 1:ns
        %Define the kernel density estimate
        KDE(i,e) = fKernelEstimateNorm(Data{i}(:,e));
    end
end


%For each component state
Pn = zeros(ns,ne);
```

```
for s = 1:ns
    %For each piece of evidence
    D = Data{s};
    n = length(D);
    Lik = zeros(n,ns);
    for e = 1:ne
        %Estimate the likelihood from the KDE for each component state
        for k = 1:ns
            Lik(:,k) = fKernelDensity(D(:,e),KDE(k,e));
        end
        %Diagnose most likely mode (here, max defines largest element in rows)
        [¬,Diagnosis] = max(Lik,[],2);

        %Define the proportion of false diagnosis results
        Pn(s,e) = sum(Diagnosis ≠ s)/n;
    end
end

%Define false negative criterion FN, a row vector with respect to evidence
FN = 1/(ns−1)*sum(Pn);
```

Given the selected modes for component 1 (m_1, m_9), one could obtain the false negative criteria for each instrument using the following code

```
FN = FalseNegativeCriterion(Data(1,9))
```

After calculating the FN criteria for all pieces of evidence, one can make a decision as to which pieces of evidence to include for that component.

Offline Step 3: Obtain kernel density estimates using selected monitors

For each mode that occurs in the data, we evaluate the kernel density based on the selected evidence. Consider the variable `EvidenceSelection` which is a cell array that contains a vector of selected evidence in each cell. The kernel density estimate can be obtained as

```
nc = 4; %number of components
nm = length(Data); %number of modes in the data

%for each component
for c = 1:nc
    %obtain selected evidence
    ind = EvidenceSelection{c};

    %for each mode
    for m = 1:nm
        %estimante KDE using selected evidence
        if ¬isempty(Data{m})
            KDE(m,c) = fKernelEstimateNorm(Data{m}(:,ind));
        end
    end
end
```

Online Step 1: Calculate likelihoods for a new data point

The Bayesian diagnosis algorithm starts off with evaluating likelihoods of modes. Here however, we need to take into account different pieces of evidence selected for components. Thus, the likelihood matrix `Lik` for evidence `e` will have columns pertaining to modes and rows pertaining to components.

```
[nm,nc] = size(KDE);

%for each mode
for m = 1:nm
    %for each component
    for c = 1:nc
        %selected evidence index
        ind = EvidenceSelection{c};
        %Evaluate likelihood of mode if it appears in the data
        if ¬isempty(KDE(m,c).bwm)
           Lik(m,c) = fKernelDensity(e(ind),KDE(m,c));
        end
    end
end
```

Online Step 2: Calculate component likelihoods

The component state likelihoods are calculated by going through all the modes having that state, and using the largest likelihood. Determining whether or not a mode has the right component state requires the use of the `ModeBV` variable defined in Eqn (7.5). As an example, if we wish to figure out the modes where component 1 is equal to 0, we search through all elements of the first column in `ModesBV` and note the rows where the element is 1. The code below, performs this search for all applicable modes and all component states.

```
[nm,nc] = size(ModesBV);
%Find the number of component states for each column
mc = max(ModesBV);

ApplicableModes = cell(1,nc);
%For each component, create a new ApplicableModes matrix
for c = 1:nc
    ApplicableModes{c} = zeros(nm,mc(c));
    %For each component state, search ModesBV
    for ci = 0:mc(c)
        %A binary matrix having 1 where the mode matches component state
        ApplicableModes{c}(:,(ci+1)) = ModesBV(:,c)==ci;
    end
end
```

After the search for applicable modes has been performed, we can calculate the likelihoods for each component state by selecting the most probable likelihood out of all applicable modes.

```
LikC = cell(1,nc);
%For each component
for c = 1:nc
    %For each component state
    for ci = 1:(mc(c)+1)
        %Use the maximum likelihood of applicable modes
        aModes = find( ApplicableModes{c}(:,ci) );
        LikC{c}(ci,1) = max(Lik(aModes,c));
    end
end
```

The result `LikC` contains likelihoods for each component state under each component.

Online Step 3: Bayesian inference

For each component, there are prior probabilities for each state. The posterior can be calculated using typical Bayesian methods.

```
%For each component
for c = 1:nc
    Post{c} = LikC{c}.*Prior{c};
    Post{c} = Post{c}/sum(Post{c});
end
```

Note that for each component `c`, `LikCc` contains column vectors of likelihoods, and `Priorc` contains column vectors of prior probabilities of the same size. The mode can be diagnosed by selecting the state values of highest probability for each component.

Online Step 4: Obtain posterior for modes (Optional)

If one wishes to display the posterior probability of the modes (assuming that the states are independent), one can use Eqn (7.3), which, in MATLAB takes the following form:

```
PostM = ones(nm,1);
%For each mode
for m = 1:nm
    %Find the component state indices
    %because zero is first index, (index = value +1)
    CompInd = ModeBV(m,:)+1;
    %For each component
    for c = 1:nc
        %Product of applicable posterior probabilities
        PostM(m) = PostM(m)*Post{c}(CompInd(c));
    end
end
```

7.3.2 Bootstrapping for additional modes

Here we discuss the technique of bootstrapping for more data. Because the end-goal is to produce more learning data, this entire technique is performed offline. If combining the two techniques, the bootstrapping procedure is completed before the first step is taken in the component space approach.

Step 1: Identify the model

The original model is defined in MATLAB according to the function `Tanks`, which yields the differential dX given the input flow rate U, current level X, and the parameters A_c (tank cross section area), C_1, C_2, C_3 coefficients for the drain, and B_1, B_2, original bias or scaling parameters for the flow rates.

```
function dX = Tanks(U,X,Ac,C1,C2,C3,B1,B2,Ho)
    dX(1,1) = ( B1*U(1) - C1*(X(1)+Ho)^(0.5) )/Ac;
    dX(2,1) = (-C2*(X(2)+Ho)^(0.5))/Ac;
    dX(3,1) = ( B2*U(2) - C3*(X(3)+Ho)^(0.5) )/Ac;
end
```

The level prediction is obtained using the RK45 method

```
function Y = HybridTanks(U,X0,Ac,C1,C2,C3,B1,B2,Ho,Ts)
%Restrict the values from being negative
options = odeset('NonNegative',[1,2,3]);
%u has rows for t and columns for components (in this case 2)
[nt,nu] = size(U);
X = X0;

for t = 1:nt
    u = OP(k,:);
    dXu = @(t,X) Tanks(u,X,Ac,C1,C2,C3,B1,B2,Ho);

    [~,vX] = ode45(dXu,[0,Ts],X,options);
    X = vX(end,:);
    X(X>100) = 100; %Upper limit on X
    Y(k,:) = X;
end
```

The parameter of the model `Theta = [Ac,C1,C2,C3,B1,B2,Ho]` can be identified by the optimization technique of your choice; for example, in MATLAB, one can use `fminunc`.

```
%U, Y X0 and Ts are already defined earlier
Theta0 = [Ac,C1,C2,C3,B1,B2]; %Initial Parameters

%Set up objective function
Yhat = @(Th) HybridTanks(U,X0,Th(1),Th(2),Th(3),Th(4),Th(5),Th(6),Th(7),Ts);
Objective = @(Th) sum( sum( (Yhat (Th) - Y ).^2 ) );

Theta = fminunc(Objective,Theta0);
```

The function `fminunc` is used for this purpose, but it may be desirable to include constraints on parameter values. Furthermore, C_1, C_2, C_3 could be forced to be equal, as the tank outlets are identical. This helps reduce dimensionality of the search.

Step 2: Obtain residual errors

When identifying model error covariances, it is required that the Q and R matrices are identified first. The first step is to set up the UKF so that state errors `Xe` $= \varepsilon_x$ and

observation errors `Ye` $= \varepsilon_y$ are reported.

```
function [X,P,Xe,Ye] = TankUKF(X0,Y1,U0,P0,Theta,Q,R,Ts)
%Set up differential equation model
dTank = @(t,X)Tanks(X,u,Th(1),Th(2),Th(3),Th(4),Th(5),Th(6),Th(7),Ts);

%Set up predictor for differential equation model
options = odeset('NonNegative',[1,2,3],'InitialStep',(Ts/10));
function Xf = Ftank(X,Ts)
    X = X.*(X>0);
    [¬,vX] = ode45(dTank,[0,Ts],X,options);
    Xf = (vX(end,:))';
end

%state transition and observation functions
ftank = @(X) Ftank(X,Ts);
htank = @(X) X;

%UKF result
[X,P] = ukf(ftank,X0,P0,htank,Y1,Q,R);

%Residual error in X
Xe = X-ftank(X0);

%Residual Error in Y
Ye = Y1-htank(X);
```

Now that we have a function to define prediction errors in X and Y, we can iteratively estimate Q and R

```
%Y, U, Ts and Theta are already defined
%Initial values for $Q$ and $R$ are already chosen
[nt,¬] = size(U);
X = Y(:,1);
P = Q;

%Obtain covariance constant for x and y log likelihoods
K = -0.5*(  log( (2*pi)^(length(Q))*det(Q) ) + log( (2*pi)^(length(R))*det(R) ) );

%Cholesky decomposition for easy inversion
Qr = chol(Q);
Rr = chol(R);

LogLik0 = -1/0;
dLog = 100;

%Iterate the estimation of Q and R until log likelihood converges
while dLog > 1e-5
    LogLik1 = 0;
    for t = 2:nt
        [X,P,xe,ye] = TankUKF(X,Y(t,:),U(t-1,:),P,Theta,Q,R,Ts);
        Xe(t,:) = xe;
        Ye(t,:) = ye;

        %Use a more accuate expression for inversion
        %  sum((Xe\Qr).^2) = Xe*inv(Qr)*Xe'
        LogLikX = -0.5*sum((Xe(t,:)\Qr).^2);
        LogLikY = -0.5*sum((Ye(t,:)\Rr).^2);
        LogLik1 = LogLik1 + K + LogLikX + LogLikY;
    end
```

```
31      Q = cov(Xe);
32      R = cov(Ye);
33      dLog = LogLik1 - LogLik0;
34  end
```

This procedure not only estimates covariances Q and R, but also the residual errors $\varepsilon_x = \texttt{Xe}$ and $\varepsilon_y = \texttt{Ye}$ can be obtained from the final iteration.

Step 3: Whitening residual errors

Residual error whitening can be performed by estimating an AR model, and applying it to the data. There are many techniques that can be used to estimate an AR model, and MATLAB has a command `Model = ar(y,n)` strictly for this purpose. It can be applied to all elements of the noise in X and Y as follows

```
1  %n is your selected model order
2  for i = 1:length(Q)
3      ARx{i} = ar(Xe(:,i),n);
4  end
5  for i = 1:length(R)
6      ARy{i} = ar(Ye(:,i),n);
7  end
```

After the model has been learned, it is possible to whiten the residual errors by applying the model. First, we want to create a data object so that

$$\boldsymbol{\varepsilon} = [\varepsilon_t, \varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-n}]$$

This can be done using the following code

```
1  for k = 1:length(Q)
2      %Construct the desired noise sequence for each residual sequence k
3      XEk = zeros(length(Xe(:,k))-(n-1),n);
4      %For each coefficient of A
5      for i = 1:n
6          xe = Xe(:,k);
7          %remove the first n - i data points
8          xe(1:(n-i)) = [];
9          %remove the last i-1 data points
10         xe((end-i+2):end) = [];
11         %Place the result in the kth XE matrix
12         XEk(:,i) = xe;
13     end
14     XE{k} = XEk;
15 end
```

The output of the AR modeling step in MATLAB is such that

$$\begin{aligned}\varepsilon_t^w &= A(1)\varepsilon_t + A(2)\varepsilon_{t-1} + \ldots + A(n)\varepsilon_{t-(n-1)} \\ &= \boldsymbol{\varepsilon}_t \times A\end{aligned}$$

In MATLAB, given the `XE` variable, the whitened output is obtained as

```
%For each residual error sequence
for k = 1:length(Q)
    A = (ARx{k}.a)';
    Xew(:,k) = XE{k}*A;
end
```

The final product `Xew` contains the whitened residuals of `Xe`. This procedure is repeated for residual error sequences in Y as well.

Step 4: Bootstrapping residual errors

When running a new simulation, new residual error sequences must be generated. This can be done by first obtaining the kernel density estimate

```
%For each X residual error component
for k = 1:length(Q)
    KDE_xew(k) = fKernelEstimateNorm(Xew(:,k));
end

%For each Y residual error component
for k = 1:length(R)
    KDE_yew(k) = fKernelEstimateNorm(Yew(:,k));
end
```

Then, data points in the history can be randomly selected

```
%nt is the number of time samples desired in the simulation
XewBS = zeros(nt,length(Q));
%For each X residual error component
for k = 1:length(Q)
    %Select data from KDE object
    Data = KDE_xew(k).data;
    nd = length(Data);
    %For as many time instances desired
    for t = 1:nt
        %Select random index from data
        ind = round(nd*rand(1));
        XewBS(t,k) = Data(ind,:);
    end
end
```

If these data points were directly used to simulate the noise, the process would be regular bootstrapping which samples directly from the historical data. Smoothed bootstrapping on the other hand, samples from the kernel density estimate, which can be done by first selecting the random piece of historical data (as done in bootstrapping) and then adding noise that's sampled from the kernel function. For the Gaussian kernel, one simply samples from the standard normal distribution, and multiplies the result by the square-root of the bandwidth. The result is a smoothed bootstrap sample, which is obtained by implementing the following MATLAB code.

```
for k = 1:length(Q)
    %Select bandwidth matrix from KDE object
    H = KDE_xew(k).bwm;
    %Smooth the bootstrap with Gaussian noise
    XewBS(:,k) = XewBS(:,k) + randn(nt,1)*H^0.5;
end
```

The smoothed bootstrapping procedure is performed on residual errors in X above, but the process should also be repeated in residual errors on Y.

Finally, one has to reverse the AR model to predict a new sequence. The reverse model takes the form

$$A(1)\varepsilon_t = -A(2)\varepsilon_{t-1} - \ldots - A(n)\varepsilon_{t-(n-1)} + \varepsilon_t^w$$

where $A(1) = 1$ by convention. This reverse AR model can be applied using the following MATLAB code

```
XeBS = zeros(nt,length(Q));
%For each component
for k = 1:length(Q)
    %Obtain AR parameters and set up noise sequence
    A = ARx{k}.a;
    na = length(A);
    y = zeros(nt+na,1);

    %Reverse model: y = -A(2) y-1 - ... - A(n) y-(n-1) + XewBS
    %Time starts at 1+length(A), as A pertains to past values
    for t = ((1:nt)+na)
        %Select past inputs for reverse model
        %wrev reverses vector (we can only construct ascending)
        ind = wrev( (t-(na-1)):(t-1) );
        %Predict new output from reverse model
        y(t) = -A(2:end)*y(ind) + XewBS(t-na);
    end
    %remove the first na inputs as they are not used
    y(1:na) = [];
    XeBs(k,:) = y;
end
```

This reverse modelling procedure should likewise be performed on residual errors in Y in addition to X which was performed above.

Step 5: Simulating new data from the bootstrap

Once the bootstrapped noise sequence has been generated, this sequence has to be inserted into a simulation. The simulation itself must contain all aspects of the control loop. For the hybrid tank system, the control loop is shown in Figure 7.3. In this model, the appropriate places to add the noise are indicated, as well as the appropriate places to sample outputs. Simulation can be done using MATLAB code, Simulink or any other desired simulation software.

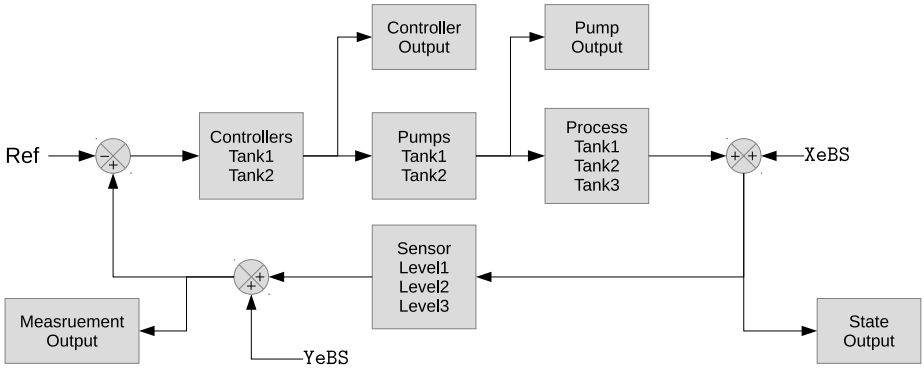


Figure 7.3: Hybrid tank control system

Step 6: Generate new data from monitoring

From the simulated data, monitoring algorithms can be applied. In this case, the monitors include an augmented unscented Kalman filter for the four components (two sensors and two valves) and a pump prediction model (for the two sensors).

7.4 Application

In this chapter, only the hybrid tank system will be considered for the following reasons

1. The model is simple enough for most audiences to grasp the nuances of the bootstrapping technique.

2. The monitors for this system have already been explained, so that the monitor selection results for this system will be most meaningful.

3. Unlike the industrial system, we have control over what modes appear in the data so that we can easily validate performance for modes that do not appear in the historical data.

Experimental data was obtained for the 16 possible modes, roughly one hour's worth of data for each mode. The nonlinear model was estimated using the Prediction-Error method, based on the lab data containing no faults. The closed-loop behaviour was then simulated for each of the 16 possible modes.

The first purpose of this experiment is to assess how the bootstrap simulation approach performs. The second purpose is to assess how well the mode-space method and component-space method are improved by replacing simulated data with real process data. For the mode-space method, replacing simulated data with process data for a single mode will improve the result for that mode, resulting in a significant but localized improvement. Conversely, for the component-space method, the same data replacement for the same

mode should improve results in diagnosing all components, resulting in a slight but more widespread improvement. For realism, real data is only applied to the most common modes, namely, normal operation and modes with a single fault occurrence.

Because the system has already been modelled, the mode space and component space methods are compared to another intuitive method, the model-based approach. The model-based method functions similarly to the component-space method, except only the corresponding UKF fault parameter estimate is used to diagnose the fault. If the UKF reading for fault parameters exceeds a certain boundary, the corresponding fault is considered present.

7.4.1 Monitor selection

When setting up the component space diagnosis approach, the monitors had to be selected based on their sensitivity toward that component's faulty state. It was found that leaks did not affect the pump-based bias monitors, thus these monitors were discarded when diagnosing bias. Additionally, because of the complex interactions between bias and leaks when implmenting the UKF, both bias and leak parameters were used for Tank 1 when diagnosing leaks and bias in that tank. Likewise, both UKF parameters are included when diagnosing bias and leaks in Tank 2. A summary of monitor inclusion is given in Table 7.1

Table 7.1: Included monitors for component space appraoch

	Bias Tank 1	Bias Tank 2	Leak Tank 1	Leak Tank 2
UKF B_1	yes	no	yes	no
UKF B_2	no	yes	no	yes
UKF L_1	yes	no	yes	no
UKF L_2	no	yes	no	yes
Pump B_1	yes	no	no	no
Pump B_2	no	yes	no	no

7.4.2 Component diagnosis

The three diagnosis methods were tested using monitor results obtained from the laboratory setup. For the first trial, diagnosis was performed using only simulated data for training. For the second trial, diagnosis was performed using experimental data from the normal operation. Finally, for the third trial, experimental data was included from all modes with a single fault. In all runs, each method attempted to diagnose the mode and the individual faults. Mode diagnosis results are shown in Table 7.2 and component diagnosis results are shown in Table 7.2

One of the first observations that can be made is that it was more difficult to diagnose the mode than it was to diagnose the individual component faults. This is expected, as the mode

Table 7.2: Misdiagnosis rates for modes

	all simulated	real data: $n_f \leq 0$	real data: $n_f \leq 0$
model driven	42.7 %	38.5 %	30.3 %
mode space	50.3 %	40.3 %	22.1 %
component space	42.6 %	37.9 %	23.1 %

Table 7.3: Misdiagnosis rates for component faults

	all simulated	real data: $n_f \leq 0$	real data: $n_f \leq 1$
model driven	13.4 %	12.6 %	9.12 %
mode space	20.1 %	15.8 %	8.9 %
component space	12.3 %	11.3 %	6.6 %

will be considered as misdiagnosed even if all components are correctly diagnosed except one. For example, if the three out of four faults were correctly diagnosed, the misdiagnosis of a single fault would mean that the entire mode is misdiagnosed. Misdiagnosis rates are therefore expected to be two to four times higher for modes than the individual faults.

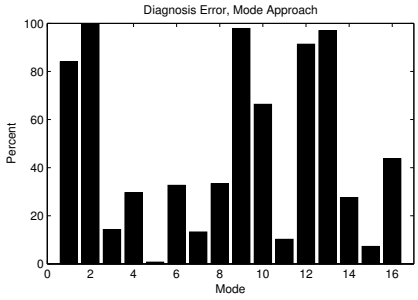
It was also found that the performance of the mode space method was inferior to that of the component space method. This is best explained by the fact that diagnosing based on component space was simpler than diagnosing based on modes. There are fewer distributions to estimate when diagnosing the presence of each fault. Furthermore, monitors not sensitive to the fault can be discarded, further reducing dimensionality of the distributions to be estimated.

While the component-based method has its merits in this application, we should note that this method assumes that component states are independent of each other. That assumption holds true for this system as the leaks and bias were introduced independently. The mode space approach does not make such assumptions; thus, if sufficient data is available for all modes, the mode space method becomes more practical, as it can better take into account interactions between states of different components.

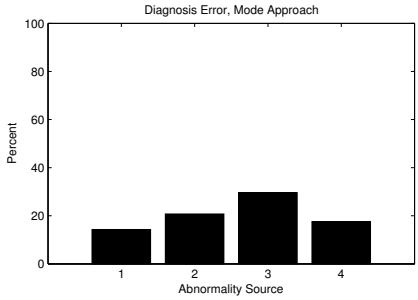
Adding experimental data to the learning dataset (first the normal mode, and then single fault modes) was found to improve performance, especially the mode space approach. Improvements from adding experimental data indicated that the model was not completely able to replicate the process; however, simulated data did provide valuable information for diagnosis as performance was still acceptable when only simulated data was used as a reference. Adding experimental data had a more balanced effect when it was introduced in the component space approach. Every time experimental data was included for a mode, half

of the distributions in the component space approach were effected, but only one distribution in the mode space was affected. Thus, the component space approach had a more evenly spread benefit.

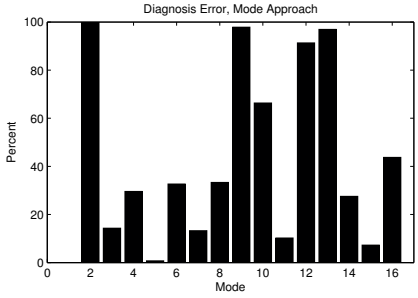
These trends can be observed in Figure 7.5 where including the normal operating mode (m_1) decreased its misdiagnosis rate, but left other modes fairly untouched. This is also true when experimental data from single fault modes (m_2, m_3, m_5, m_9) was added; misdiagnosis rates for these modes decreased, but the other modes were left untouched. Conversely, when observing Figure 7.4, including experimental data from these modes affected the misdiagnosis rate of all modes, usually resulting in a decreased misdiagnosis rate for modes on average.



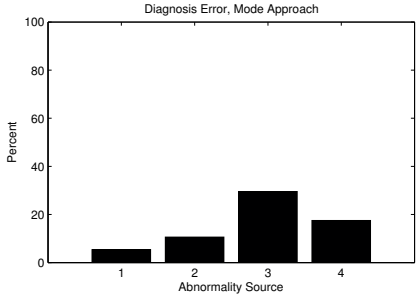
(a) No lab data: 50.3 % error



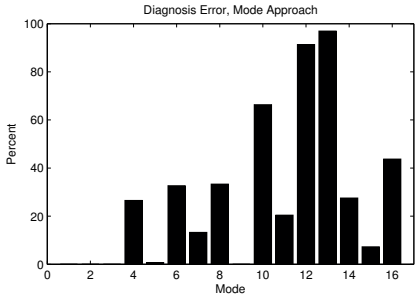
(b) No lab data: 20.1 % error



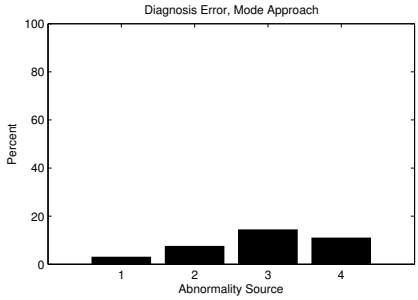
(c) Normal mode lab data: 40.3 % error



(d) Normal mode lab data: 15.8 % error

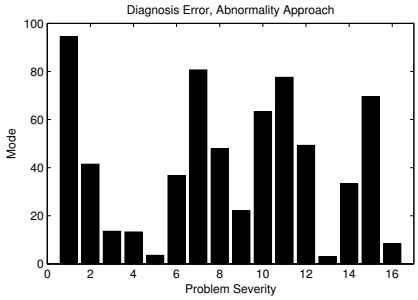


(e) Single fault lab data: 22.1 % error

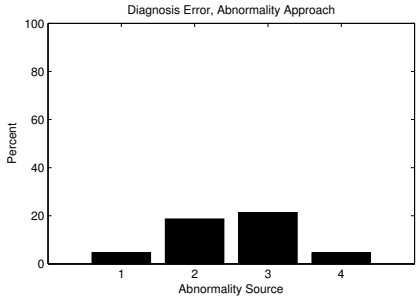


(f) Single fault lab data: 8.9 % % error

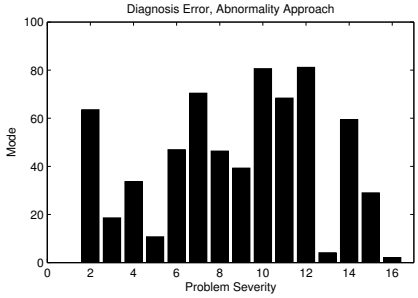
Figure 7.4: Diagnosis results for mode space approach



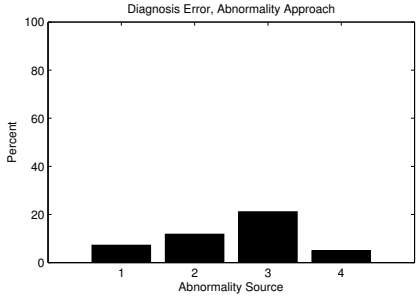
(a) No lab data: 42.6 % error



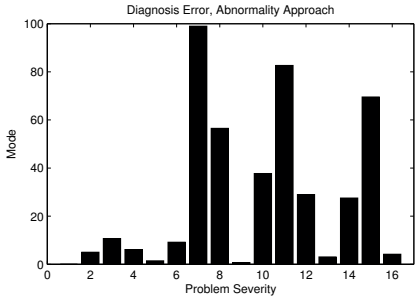
(b) No lab data: 20.1 % error



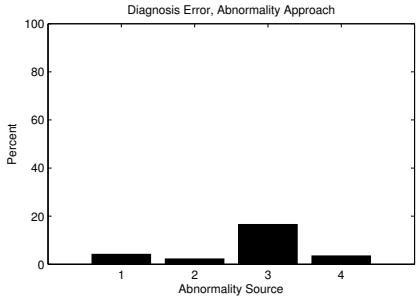
(c) Normal mode lab data: 37.9 % error



(d) Normal mode lab data: 12.3 % error



(e) Single fault lab data: 23.1 % error



(f) Single fault lab data: 6.8 % % error

Figure 7.5: Diagnosis results for component space approach

Part II

Application

Chapter 8

Accounting for ambiguous modes in historical data: A Bayesian approach

8.1 Introduction

This chapter discusses the application of the second-order parametrization method, which is useful when the historical data contains ambiguous operating modes (i.e. uncertain cases where more than one mode is possible). Ambiguity occurs when information is missing from one or more problem sources. For example, let us say that there are two components, an sensor and a valve. The sensor can develop bias, while the valve can become sticky. If bias is known to exist, but it is unknown whether the valve is sticky or not, then there are two possible modes, one mode having bias without stiction, and the other having bias with stiction.

When ambiguous modes are in the data, the resulting probabilities can have ranges. The second-order method is used to combine prior probabilities along with one or more likelihoods in order to obtain a final result with probability boundaries. The approach consists of the following steps:

1. Set up a method for calculating likelihoods given θ: $p(E|M, \Theta))$
2. Calculate second-order approximation of likelihood expression
3. Combine likelihood expressions with the second-order combination rule to obtain final diagnosis result
4. An option to group monitors together into approximately independent groups

8.2 Algorithm

8.2.1 Formulating the problem

When ambiguous modes $\boldsymbol{M}$ are present in the data, the likelihood expression takes the following form:

$$p(E|M,\Theta) = \frac{S(E|M)n(M) + \sum\limits_{\boldsymbol{m}_k \supset M} \theta\{\frac{M}{\boldsymbol{m}_k}\} S(E|\boldsymbol{m}_k) n(\boldsymbol{m}_k)}{n(M) + \sum\limits_{\boldsymbol{m}_k \supset M} \theta\{\frac{M}{\boldsymbol{m}_k}\} n(\boldsymbol{m}_k)} \tag{8.1}$$

where $n(\boldsymbol{m}_k)$ is the number of times $\boldsymbol{m}_k$ appears in the history, and $S(E|\boldsymbol{m}_k)$ is the support function for evidence E given $\boldsymbol{m}_k$; here, the term $S(E|\boldsymbol{m}_k)$ is calculated in the same manner as likelihood

$$p(E|M) = \frac{n(E|M)}{n(M)}$$
$$S(E|\boldsymbol{M}) = \frac{n(E|\boldsymbol{M})}{n(\boldsymbol{M})}$$

The only difference from the likelihood is that $\boldsymbol{m}_k$ can be an ambiguous mode. The terms $\theta\{\frac{M}{\boldsymbol{m}_k}\}$ are unknown and represent the proportions of data in an ambiguous mode $\boldsymbol{m}_k$ which belongs to one of its specific modes $M \subset \boldsymbol{m}_k$.

$$\theta\{\tfrac{M}{\boldsymbol{m}_k}\} = \frac{n(M|\boldsymbol{m}_k)}{n(\boldsymbol{m}_k)} = p(M|\boldsymbol{m}_k)$$

The unknown parameters $\theta\{\frac{M}{\boldsymbol{m}_k}\}$ can be used to define the probability boundaries, namely, the plausibility (the maximum probability) and the belief (minimum probability)

$$Bel(E|M) = \min_{\Theta} p(E|M,\Theta)$$
$$Pl(E|M) = \max_{\Theta} p(E|M,\Theta)$$

The challenge with the expression in Eqn (8.1) is that it is difficult to use Bayesian methods to combine likelihoods with priors, as the resulting expressions are increasing in complexity with respect to Θ (where Θ represents the collection of all θ parameters). In order to manage this complexity, the second-order Taylor Series approximation is used. When this approximation is applied, a simple updating rule can be derived and the resulting belief and plausibility are relatively easy to obtain.

8.2.2 Second-Order Taylor series approximation of $p(E|M,\Theta)$

The Taylor series approximation makes use of differentiation in order to make an approximation around a reference point $\hat{\Theta}$. The univariate expression is given by

$$f(x) \approx f(\hat{x}) + \left.\frac{d\ f(x)}{d\ x}\right|_{\hat{x}} (x - \hat{x}) + \frac{1}{2} \left.\frac{d^2\ f(x)}{d\ x^2}\right|_{\hat{x}} (x - \hat{x})^2 + \ldots \tag{8.2}$$

where $\hat{x}$ is a reference point around which the approximation is centred. However, the expression $p(E|M,\Theta)$ is a function of multiple variables Θ, thus a multivariate Taylor series approximation is needed. In this case, a second-order multivariate Taylor Series approximation is desired:

$$\begin{aligned}\Delta\Theta &= \hat{\Theta} - \Theta \\ p(E|M,\Theta) &= p(E|M,\hat{\Theta}) + \boldsymbol{J}\Delta\Theta + \frac{1}{2}\Delta\Theta^T\boldsymbol{H}\Delta\Theta \end{aligned} \tag{8.3}$$

where $\boldsymbol{J}$ and $\boldsymbol{H}$ are Jacobian and Hessian matrices

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial\ p(E|M,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m_1}}\}} & \frac{\partial\ p(E|M,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m_2}}\}} & \cdots & \frac{\partial\ p(E|M,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m}_N}\}} \end{bmatrix}$$

$$\boldsymbol{H} = \begin{bmatrix} \frac{\partial^2\ p(E|M,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m}_1}\}^2} & \frac{\partial^2\ p(E|M,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m}_1}\}\theta\{\frac{M}{\boldsymbol{m}_2}\}} & \cdots & \frac{\partial^2\ p(E|M,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m}_1}\}\theta\{\frac{M}{\boldsymbol{m}_N}\}} \\ \frac{\partial^2\ p(E|M,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m}_2}\}\partial\ \theta\{\frac{M}{\boldsymbol{m}_1}\}} & \frac{\partial^2\ p(E|M,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m}_2}\}^2} & \cdots & \frac{\partial^2\ p(E|M,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m}_2}\}\theta\{\frac{M}{\boldsymbol{m}_N}\}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\ p(E|M,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m}_N}\}\partial\ \theta\{\frac{M}{\boldsymbol{m}_1}\}} & \frac{\partial^2\ p(E|M,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m}_N}\}\theta\{\frac{M}{\boldsymbol{m}_2}\}} & \cdots & \frac{\partial^2\ p(E|M,\Theta)}{\partial\ \theta\{\frac{M}{\boldsymbol{m}_N}\}^2} \end{bmatrix}$$

The expressions for the partial derivatives with respect to $p(E|M,\Theta)$ are obtained by differentiating Eqn (8.1). For compactness of notation, we introduce $\boldsymbol{S}$ and $\boldsymbol{n}$ and $\boldsymbol{\theta}$ as vectors.

$$\begin{aligned}\boldsymbol{S} &= [S(E|\boldsymbol{m}_1), S(E|\boldsymbol{m}_1), \ldots, S(E|\boldsymbol{m}_n)] \\ \boldsymbol{n} &= [n(\boldsymbol{m}_1), n(\boldsymbol{m}_1), \ldots, n(\boldsymbol{m}_n)] \\ \boldsymbol{\theta} &= \left[\theta\{\tfrac{M}{\boldsymbol{m}_1}\}, \theta\{\tfrac{M}{\boldsymbol{m}_2}\}, \ldots, \theta\{\tfrac{M}{\boldsymbol{m}_n}\}\right]\end{aligned}$$

The partial differentials are then given as

$$\begin{aligned}\frac{\partial\ p(E|M,\Theta)}{\partial\ \boldsymbol{\theta}_i} &= \frac{\boldsymbol{n}_i\boldsymbol{S}_i}{\sum_k \boldsymbol{n}_k\boldsymbol{\theta}_k} - \frac{\boldsymbol{n}_i\ \sum_k \boldsymbol{S}_k\boldsymbol{n}_k\boldsymbol{\theta}_k}{\left(\sum_k \boldsymbol{n}_k\boldsymbol{\theta}_k\right)^2} \\ \frac{\partial^2\ p(E|M,\Theta)}{\partial\ \boldsymbol{\theta}_i\ \partial\ \boldsymbol{\theta}_j} &= -\frac{\boldsymbol{n}_i\boldsymbol{S}_j + \boldsymbol{n}_j\boldsymbol{S}_i}{\left(\sum_k \boldsymbol{n}_k\boldsymbol{\theta}_k\right)^2} + \frac{\boldsymbol{n}_i\boldsymbol{n}_j\ \sum_k \boldsymbol{S}_k\boldsymbol{n}_k\boldsymbol{\theta}_k}{\left(\sum_k \boldsymbol{n}_k\boldsymbol{\theta}_k\right)^3}\end{aligned}$$

Reference point: The informed transformation

The second-order Taylor series approximation requires a reference point for Θ in order to obtain expressions for each likelihood. The informed transformation makes use of the most credible assignment of Θ as a reference point based on prior probabilities. As an example, let us consider a three-mode system with the following priors: $p(m_1) = 0.5$, $p(m_2) = 0.25$ and $p(m_3) = 0.25$. Now let us say that there is a body of evidence belonging to the ambiguous mode $\{m_1, m_2\}$. If, according to the prior probabilities, mode m_1 was twice as probable as

mode m_2 then the most credible allocation is

$$\hat{\theta}\{\tfrac{m_1}{\{m_1,m_2\}}\} = p(m_1|\{m_1,m_2\}) = \frac{0.5}{0.5+0.25} = 2/3$$
$$\hat{\theta}\{\tfrac{m_2}{\{m_1,m_2\}}\} = p(m_2|\{m_1,m_2\}) = \frac{0.25}{0.5+0.25} = 1/3$$

where $\hat{\theta}$ indicates a most credible estimate. Similarly, consider body of evidence belonging to the ambiguous mode $\{m_2, m_3\}$ where the priors of m_2 and m_3 are equal. The most credible allocation in this case is

$$\hat{\theta}\{\tfrac{m_2}{\{m_2,m_3\}}\} = p(m_2|\{m_2,m_3\}) = \frac{0.25}{0.25+0.25} = 1/2$$
$$\hat{\theta}\{\tfrac{m_3}{\{m_2,m_3\}}\} = p(m_3|\{m_2,m_3\}) = \frac{0.25}{0.25+0.25} = 1/2$$

This technique to obtain credible allocation is called the *informed transformation* as it is based on information from prior probabilities. In general, the informed transformation $\hat{\theta}\{\frac{M}{\boldsymbol{m}}\}$ is given as

$$\hat{\theta}\{\tfrac{M}{\boldsymbol{m}_k}\} = \frac{p(M)}{\sum\limits_{M\subset\boldsymbol{m}_k} p(M)} \tag{8.4}$$

This transformation also yields the informed likelihood $\hat{p}(E|M)$ by substituting $\hat{\Theta}$ in for Θ in the likelihood expression in Eqn (8.1)

$$\hat{p}(E|M) = p(E|M,\hat{\Theta})$$

8.2.3 Second-Order Bayesian combination

Combination method

Two types of combination need to be considered: the combination of multiple likelihoods and combination with a prior. In both cases, two second-order approximations are multiplied together and all terms that are third order or higher are discarded. The difference lies in the normalization. For combining likelihoods, normalization is not needed, but when combining with a prior using Bayes' Rule, normalization must be performed.

The goal for likelihood combination is to express the joint probability of two independent pieces of evidence.

$$p(E_1,E_2|M,\Theta) = p(E_1|M,\Theta)p(E_2|M,\Theta)$$
$$p(E_1|M,\Theta) \approx \hat{p}(E_1|M) + \boldsymbol{J_1}\Delta\Theta + \frac{1}{2}\Delta\Theta^T\boldsymbol{H_1}\Delta\Theta$$
$$p(E_2|M,\Theta) \approx \hat{p}(E_2|M) + \boldsymbol{J_2}\Delta\Theta + \frac{1}{2}\Delta\Theta^T\boldsymbol{H_2}\Delta\Theta$$

When ignoring terms of third order and higher, the joint probability must also be expressed as second-order

$$p(E_1,E_2|M,\Theta) \approx \hat{p}(E_1,E_2|M) + \boldsymbol{J_{12}}\Delta\Theta + \frac{1}{2}\Delta\Theta^T\boldsymbol{H_{12}}\Delta\Theta$$

By multiplying the second order approximations of $p(E_1|M,\Theta)$ and $p(E_2|M,\Theta)$ and then discarding terms that are third order and higher, one obtains a rule for updating key terms

$$\begin{aligned}\hat{p}(E_1,E_2|M) &= \hat{p}(E_1|M)\hat{p}(E_2|M)\\ \boldsymbol{J_{12}} &= \hat{p}(E_1|M)\boldsymbol{J_2} + \hat{p}(E_2|M)\boldsymbol{J_1}\\ \boldsymbol{H_{12}} &= \hat{p}(E_1|M)\boldsymbol{H_2} + \hat{p}(E_2|M)\boldsymbol{H_1} + \boldsymbol{J_2}^T\boldsymbol{J_1} + \boldsymbol{J_1}^T\boldsymbol{J_2}\end{aligned}$$

After the independent likelihoods are combined, they can be combined with a prior distribution $p(M|\Theta)$ in a similar manner in order to obtain the posterior estimate $p(M|E,\Theta)$. However, for combination with a prior, the normalization constant K needs to be accounted for. First, we define the prior

$$p(M|\Theta) = \hat{p}(M) + \boldsymbol{J_P}\Delta\Theta + \frac{1}{2}\Delta\Theta^T\boldsymbol{H_P}\Delta\Theta$$

In the static case, there is often no ambiguity in the prior, and in such a case $\boldsymbol{J_P}$ and $\boldsymbol{H_P}$ are zero matrices. When combining the likelihood with the prior, normalization is required

$$K = \sum_k \hat{p}(m_k)\hat{p}(E_1,E_2|m_k)$$

The second-order terms for the posterior probability are similar to the likelihood, except now the terms are normalized by the inverse of K

$$\begin{aligned}\hat{p}(M|E_1,E_2) &= \frac{1}{K}\hat{p}(m_k)\hat{p}(E_1,E_2|m_k)\\ \boldsymbol{J_F} &= \frac{1}{K}\left[\hat{p}(M)\boldsymbol{J_{12}} + \hat{p}(E_1,E_2|M)\boldsymbol{J_P}\right]\\ \boldsymbol{H_F} &= \frac{1}{K}\left[\hat{p}(M)\boldsymbol{H_{12}} + \hat{p}(E_1,E_2|M)\boldsymbol{H_P} + \boldsymbol{J_{12}}^T\boldsymbol{J_P} + \boldsymbol{J_P}^T\boldsymbol{J_{12}}\right]\end{aligned}$$

so that the second-order expression for the posterior probability is

$$p(M|E_1,E_2,\Theta) = \hat{p}(M|E_1,E_2) + \boldsymbol{J_F}\Delta\Theta + \frac{1}{2}\Delta\Theta^T\boldsymbol{H_F}\Delta\Theta$$

Diagnosis methods: The simple method

The diagnosis is made by determining which posterior probability has the highest value. A simple posterior probability estimate is the second-order reference point, $\hat{p}(M|E_1,E_2)$ (or the informed transformation). This value is already given in the second-order probability calculations and requires no information from $\boldsymbol{J}$ or $\boldsymbol{H}$. In fact, if the objective is to make a diagnosis based on the simple point estimate, $\boldsymbol{J}$ or $\boldsymbol{H}$ do not need to be calculated at all. One can simply use the Bayesian method based on the reference likelihoods $\hat{p}(E_1,E_2|M)$.

Diagnosis methods: The expected value method

The more complex and rigorous estimate is the expected value $E[p(M|E)]$ which assumes a distribution over the θ parameters and calculates the expected value of the posterior. The posterior expected value is given as

$$E[p(M|E)] = \boldsymbol{C} + \boldsymbol{J}^* E[\Theta] + \frac{1}{2}\left[E[\Theta]^T \boldsymbol{H_{OD}} E[\Theta] + E[\Theta^2]^T \boldsymbol{H_D}\right] \tag{8.5}$$

where $\boldsymbol{H_{OD}}$ is the Hessian $\boldsymbol{H}$ with the diagonal elements being set to zero, and $\boldsymbol{H_D}$ is a column vector of the diagonal elements of $\boldsymbol{H}$. The parameters $\boldsymbol{C}$ and $\boldsymbol{J}^*$ are calculated as

$$\begin{aligned}\boldsymbol{C} &= \hat{p}(M|E) - \boldsymbol{J}\hat{\Theta} + \frac{1}{2}\hat{\Theta}^T \boldsymbol{H}\hat{\Theta} \\ \boldsymbol{J}^* &= (\boldsymbol{J} - \hat{\Theta}^T \boldsymbol{H})\end{aligned}$$

where $\hat{\Theta}$ is the same value as that given in Eqn (8.4). The terms $E[\Theta]$ and $E[\Theta^2]$ are column vectors with each element pertaining to $E[\theta\{\frac{M}{\boldsymbol{m}_k}\}]$ and $E[\theta^2\{\frac{M}{\boldsymbol{m}_k}\}]$. These expectations are calculated assuming that θ follows a Dirichlet distribution; for each term, the expectation results are given as follows:

$$E[\theta\{\tfrac{M}{\boldsymbol{m}_k}\}] = \frac{\alpha(\frac{M}{\boldsymbol{m}_k})}{\sum\limits_{m_i \subset \boldsymbol{m}_k} \alpha(\frac{m_i}{\boldsymbol{m}_k})} \tag{8.6}$$

$$E[\theta^2\{\tfrac{M}{\boldsymbol{m}_k}\}] = \frac{\alpha(\frac{M}{\boldsymbol{m}_k}) + \alpha^2(\frac{M}{\boldsymbol{m}_k})}{\left[\sum\limits_{m_i \subset \boldsymbol{m}_k} \alpha(\frac{m_i}{\boldsymbol{m}_k})\right] + \left[\sum\limits_{m_i \subset \boldsymbol{m}_k} \alpha(\frac{m_i}{\boldsymbol{m}_k})\right]^2} \tag{8.7}$$

where $\alpha(\frac{M}{\boldsymbol{m}_k})$ represents the prior frequency of mode M happening given the ambiguous mode $\boldsymbol{m}_k$. A natural value for $\alpha(\frac{M}{\boldsymbol{m}_k})$ can be calculated as

$$\alpha(\tfrac{M}{\boldsymbol{m}_k}) = p(M|\boldsymbol{m}_k)n(\boldsymbol{m}_k) = \hat{\theta}\{\tfrac{M}{\boldsymbol{m}_k}\}n(\boldsymbol{m}_k)$$

Diagnosing by means of the expected value is advantageous when $n(\boldsymbol{m}_k)$ is small (hence, historical data with a large number of ambiguous modes and few data points). Small values of $n(\boldsymbol{m}_k)$ lead to large variances of the Θ parameters. If there is a large number of data points for an ambiguous mode, the variances of Θ will be much smaller, and the expected values will be nearly identical to the much simpler point estimates obtained from the informed transformation.

8.2.4 Optional step: Separating monitors into independent groups

Overview of motivation

If the evidence E is multivariate with a large number of components, the likelihood function $p(E|M)$ is high-dimensional and can be difficult to estimate, so it is often desirable to break down E into independent groups. For example, if E has ten elements, and each element can take on two values, then the distribution would be ten-dimensional with $2^{10} = 1024$ different possible values (resulting in a 1042 bin distribution). However, if elements in E can be divided into two independent groups (E_1 and E_2) with five elements each, we would end up estimating two five dimensional distributions with $2^5 = 32$ bins each. A

significant simplification over the former 1024-bin distribution. The joint probability could be calculated through simple multiplication

$$p(E_1, E_2|M) = p(E_1|M)p(E_2|M) \qquad E_1|m \perp E_2|m$$

Note that evidence groups can change with each mode, so they only have to be conditionally independent given the mode. This results in the following Bayesian combination rule

$$\begin{aligned} p(M|E_1, E_2) &= \frac{p(E_1, E_2|M)p(M)}{\sum_k p(E_1, E_2|m_k)p(m_k)} \\ &= \frac{p(E_1|M)p(E_2|M)p(M)}{\sum_k p(E_1, E_2|m_k)p(m_k)} \end{aligned}$$

where $p(E_1, E_2|m_k)$ can be broken down differently depending on m_k. Breaking down likelihoods into components gives added utility to the second-order combination rule; when ambiguous modes are present, the second-order rule is needed to calculate the joint probabilities and their respective ranges.

Dependency metrics

In order to break down the likelihood into separate components, a dependency metric is required. A popular metric is the *Mutual Information Criterion* (MIC).

$$I(X_1; X_2) = \int_{\mathcal{X}_1} \int_{\mathcal{X}_2} p(x_1, x_2) \log \left(\frac{p(x_1, x_2)}{p(x_1)p(x_2)} \right) \, dx_1 \, dx_2 \tag{8.8}$$

where $\mathcal{X}_1$ and $\mathcal{X}_2$ represent the domain of the PDF $p(x_1, x_2)$. In this chapter, evidence components x_1 and x_2 are discrete, thus the MIC is calculated via summation instead of integration. The MIC yields a value of zero if X_1 and X_2 are independent, and a positive value if they are dependent (and ideally a value of infinity if they are perfectly dependent).

8.2.5 Grouping methodology

The MIC values should first be arranged in a matrix, in a similar manner to covariance. In order to avoid redundancy, we only consider the lower-left hand portion of this matrix

$$MIC(\boldsymbol{X}) = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ MIC(X_2, X_1) & 0 & 0 & \cdots & 0 \\ MIC(X_3, X_1) & MIC(X_3, X_2) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ MIC(X_p, X_1) & MIC(X_p, X_2) & MIC(X_p, X_3) & \cdots & 0 \end{bmatrix}$$

By assuming independence between groups, elements of the MIC matrix will be assumed as zero. In order to minimize the loss of information, elements of the MIC are sorted from largest to smallest. The grouping algorithm starts with the largest MIC value, which corresponds to X_a and X_b. The first group to be formed is thus $\{X_a, X_b\}$. The algorithm then proceeds to the next MIC value. For any newly drawn MIC value for $\{X_a, X_b\}$ one is faced with one of four different scenarios:

1. Scenario: Variables $\{X_a, X_b\}$ are not contained in any previous groups
   Solution: Start a new group which contains $\{X_a, X_b\}$

2. Scenario: X_a is contained in one of the previous groups but X_b is not
   Solution: Add variable X_b to the group that contains $\{X_a\}$

3. Scenario: X_a and X_b are contained in two different existing groups
   Solution: Consider merging the two existing groups together as long as the merged group is not too large

4. Scenario: X_a and X_b are both contained in the same existing group
   Solution: Do nothing, as the $MIC(X_a, X_b)$ is already taken into account

This is performed until either all MIC values are accounted for , or if the MIC values become small (for example $MIC < 0.01$.

8.3 Illustrative Example of Proposed Methodology

8.3.1 Introduction

We now go over a simple example of how to implement the proposed algorithm. Consider a control loop as shown in Figure 8.1; let us assume that the sensor may be subject to bias, and the valve may be subject to stiction. Consider two pieces of evidence in the form of monitors; E_1 is a bias monitor with outputs "bias" and "no bias", while E_2 is a stiction monitor with outputs "siction" and "no stiction". Positive results are given as (1) while negative results are given as (0).

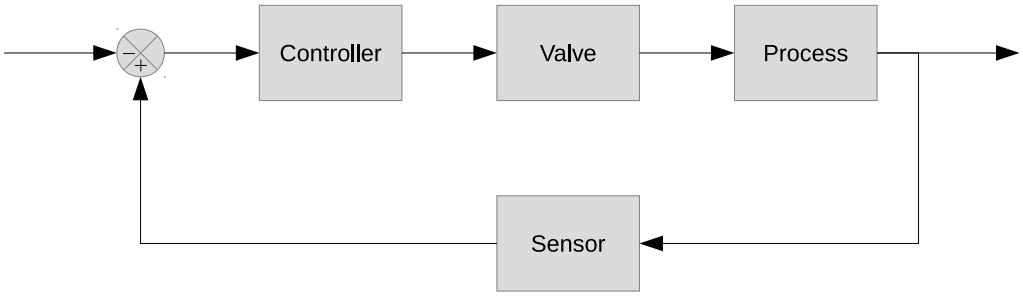


Figure 8.1: Typical control loop

8.3.2 Offline Step 1: Historical data collection

The first step is to go through the historical data and note the instances where each of the four possible modes occurs.

1. m_1 $[0, 0]$ where bias and stiction do not occur
2. m_2 $[0, 1]$ where bias does not occur but stiction does
3. m_3 $[1, 0]$ where bias occurs but stiction does not
4. m_4 $[1, 1]$ where both bias and stiction occur

Data should be collected according to each mode. In certain instances, one of these ambiguous modes may also occur

1. $\{m_1, m_3\}$ $[\times, 0]$ where bias is undetermined and stiction does not occur
2. $\{m_2, m_4\}$ $[\times, 1]$ where bias is undetermined and stiction occurs
3. $\{m_1, m_2\}$ $[0, \times]$ where bias does not occur and stiction is undetermined
4. $\{m_3, m_4\}$ $[1, \times]$ where bias occurs and stiction is undetermined
5. $\{m_1, m_2, m_3, m_4\}$ $[\times, \times]$ where both bias and stiction are undetermined

Such data should be collected according to each ambiguous mode.

8.3.3 Offline Step 2: Mutual Information Criterion (optional)

This step is optional, and considering the data is only two-dimensional, the benefit of assuming independence is insignificant in this case. However, for the sake of demonstration, the MIC matrix can be calculated as follows

$$MIC = \begin{bmatrix} 0 & 0 \\ MIC(1,2) & 0 \end{bmatrix}$$

where $MIC(1, 2)$ is calculated as

$$MIC(1,2) = \sum_{E_1} \sum_{E_2} p(E_1, E_2|m) \log \left(\frac{p(E_1, E_2|M)}{p(E_1|M)p(E_2|M)} \right)$$

The MIC matrix must be calculated for every unambiguous mode. For each mode, the two pieces of evidence are either considered independent or dependent. For example, let us say that under Mode (1), the probability is distributed as given in Table 8.1.

Table 8.1: Probability of evidence given Mode 1

	$E = [0, 0]$	$E = [0, 1]$	$E = [1, 0]$	$E = [1, 1]$
$n(E\|m_1)$	70	10	15	5
$p(E\|m_1)$	0.7	0.1	0.15	0.05

The MIC for Mode 1 is given as

$$\begin{aligned}
MIC &= p([0,0]|m_1)\log\left(\frac{p([0,0]|m_1)}{p([0,\times]|m_1)p([\times,0]|m_1)}\right) \\
&\quad + p([0,1]|m_1)\log\left(\frac{p([0,1]|m_1)}{p([0,\times]|m_1)p([\times,1]|m_1)}\right) \\
&\quad + p([1,0]|m_1)\log\left(\frac{p([1,0]|m_1)}{p([1,\times]|m_1)p([\times,0]|m_1)}\right) \\
&\quad + p([1,1]|m_1)\log\left(\frac{p([1,1]|m_1)}{p([1,\times]|m_1)p([\times,1]|m_1)}\right) \\
&= 0.7\log\left(\frac{0.7}{(0.8)(0.85)}\right) + 0.1\log\left(\frac{0.1}{(0.8)(0.15)}\right) \\
&\quad + 0.15\log\left(\frac{0.15}{(0.2)(0.85)}\right) + 0.05\log\left(\frac{0.05}{(0.2)(0.15)}\right) \\
&= 0.0088259
\end{aligned}$$

Because this number (0.0088259) is so small, the two monitors E_1 and E_2 can be considered independent under this mode.

8.3.4 Offline Step 3: Calculate reference values

The reference values $\hat{\Theta}$ can be calculated offline after the data has been collected; in particular, we work with the subsets $\hat{\boldsymbol{\theta}}\{m_i\}$ of $\hat{\Theta}$. The term $\hat{\theta}\{\frac{m_i}{m_i\subset\bullet}\}$ is defined as a vector containing all modes that can support m_i, for example,

$$\hat{\theta}\{\tfrac{m_1}{m_1\subset\bullet}\} = \left[\hat{\theta}\{\tfrac{m_1}{m_1}\}, \hat{\theta}\{\tfrac{m_1}{\{m_1,m_2\}}\}, \hat{\theta}\{\tfrac{m_1}{\{m_1,m_3\}}\}, \hat{\theta}\{\tfrac{m_1}{\{m_1,m_2,m_3,m_4\}}\}\right]^T$$
$$\hat{\theta}\{\tfrac{m_2}{m_2\subset\bullet}\} = \left[\hat{\theta}\{\tfrac{m_2}{m_2}\}, \hat{\theta}\{\tfrac{m_2}{\{m_1,m_2\}}\}, \hat{\theta}\{\tfrac{m_2}{\{m_2,m_4\}}\}, \hat{\theta}\{\tfrac{m_2}{\{m_1,m_2,m_3,m_4\}}\}\right]^T$$

Note that $\hat{\theta}\{\frac{m_1}{m_1}\} = 1$ and $\hat{\theta}\{\frac{m_2}{m_2}\} = 1$ by definition. Each element of $\hat{\theta}\{\frac{m_i}{m_i\subset\bullet}\}$ is calculated as

$$\hat{\theta}\{\tfrac{m_i}{\boldsymbol{m}_k}\} = \frac{n(m_i)}{\sum\limits_{m_j\subseteq\boldsymbol{m}_k} n(m_j)} \qquad m_i \subseteq \boldsymbol{m}_k$$

For example, let us say that the prior probabilities of the four modes are given in Table 8.2 Then parameters $\hat{\theta}\{\frac{m_i}{\boldsymbol{m}_k}\}$ can be calculated, for example, as

Table 8.2: Prior probabilities

	m_1	m_2	m_3	m_4
$p(E\|m_1)$	0.6	0.15	0.15	0.1

$$\hat{\theta}\{\tfrac{m_1}{\{m_1,m_2\}}\} = \frac{p(m_1)}{p(m_1)+p(m_2)}$$
$$= \frac{0.6}{0.6+0.15} = 0.8$$
$$\hat{\theta}\{\tfrac{m_2}{\{m_1,m_2,m_3,m_4\}}\} = \frac{p(m_2)}{p(m_1)+p(m_2)+p(m_3)+p(m_4)}$$
$$= \frac{0.15}{0.6+0.15+0.15+0.1} = 0.15$$

However, if the term in the numerator is not contained in the denominator, the value of $\hat{\theta}\{\frac{m_i}{\boldsymbol{m}_k}\}$ is zero, for example,

$$\hat{\theta}\{\tfrac{m_1}{\{m_2,m_4\}}\} = \frac{0}{p(m_2)+p(m_4)} = 0$$

This is because given the ambiguous mode $\{m_2, m_4\}$ the probability of m_1 is zero.

8.3.5 Online Step 1: Calculate support

When a new piece of evidence $[E_1, E_2]$ becomes available, we calculate the support according to

$$S(E_1, E_2|\boldsymbol{m}_k) = \frac{n(E_1, E_2|\boldsymbol{m}_k)}{n(\boldsymbol{m}_k)}$$

For example, using the values in Table 8.1, if $E = [0, 0]$ then $S(E_1, E_2|\boldsymbol{m}_1)$ would be calculated as

$$S([0, 0]|\boldsymbol{m}_1) = \frac{70}{70+10+15+5} = 0.7;$$

If E_1 and E_2 are considered independent given m_k, the support is calculated separately

$$S(E_1|\boldsymbol{m}_k) = \frac{n(E_1|\boldsymbol{m}_k)}{n(\boldsymbol{m}_k)} \qquad S(E_2|\boldsymbol{m}_k) = \frac{n(E_2|\boldsymbol{m}_k)}{n(\boldsymbol{m}_k)}$$

Using the same example, the probability given the independence assumption would be

$$S([0, \times]|\boldsymbol{m}_k) = \frac{70+10}{70+10+15+5} = 0.8$$
$$S([\times, 0]|\boldsymbol{m}_k) = \frac{70+15}{70+10+15+5} = 0.85$$

As comparison with the first result, the joint probability given the independence assumption would be

$$S([0, 0]|\boldsymbol{m}_1) = S([0, \times]|\boldsymbol{m}_k)S([\times, 0]|\boldsymbol{m}_k) = 0.8 * 0.85 = 0.68$$

which is fairly close to our original result, $S([0, 0]|\boldsymbol{m}_1) = 0.7$; thus validating the independence assumption.

8.3.6 Online Step 2: Calculate second-order terms

The second-order terms are calculated as a linearization around $\hat{\Theta}$. The required terms are

1. $\hat{p}(E|M) = p(E|M, \hat{\Theta})$: required for probability boundaries and all diagnosis methods
2. $\left.\frac{\partial\ p(E|M,\Theta)}{\partial\ \boldsymbol{\theta}_i}\right|_{\hat{\Theta}}$: required for probability boundaries and expected value diagnosis method
3. $\left.\frac{\partial^2\ p(E|M,\Theta)}{\partial\ \boldsymbol{\theta}_i\ \partial\ \boldsymbol{\theta}_j}\right|_{\hat{\Theta}}$: required for probability boundaries and expected value diagnosis method

These terms can be calculated as follows:

$$
\begin{aligned}
p(E|M, \hat{\Theta}) &= \frac{\sum_k \boldsymbol{S}_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k}{\sum_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k} \\
\left.\frac{\partial\ p(E|M,\Theta)}{\partial\ \boldsymbol{\theta}_i}\right|_{\hat{\Theta}} &= \frac{\boldsymbol{n}_i \boldsymbol{S}_i}{\sum_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k} - \frac{\boldsymbol{n}_i\ \sum_k \boldsymbol{S}_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k}{\left(\sum_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k\right)^2} \\
\left.\frac{\partial^2\ p(E|M,\Theta)}{\partial\ \boldsymbol{\theta}_i\ \partial\ \boldsymbol{\theta}_j}\right|_{\hat{\Theta}} &= -\frac{\boldsymbol{n}_i \boldsymbol{S}_j + \boldsymbol{n}_j \boldsymbol{S}_i}{\left(\sum_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k\right)^2} + \frac{\boldsymbol{n}_i \boldsymbol{n}_j\ \sum_k \boldsymbol{S}_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k\{m\}}{\left(\sum_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k\right)^3}
\end{aligned}
$$

where $\boldsymbol{S}$ and $\boldsymbol{n}$ are horizontal vectors containing the support and frequency of the modes that can support mode M (both ambiguous and unambiguous). In order to illustrate, let us consider a more complete version of Table (8.1) shown below in Tables (8.3) and (8.4)

Table 8.3: Frequency of modes containing m_1

	$E = [0,0]$	$E = [0,1]$	$E = [1,0]$	$E = [1,1]$
$n(E\|m_1)$	70	10	15	5
$n(E\|m_2)$	14	59	6	21
$n(E\|m_3)$	13	7	58	22
$n(E\|m_4)$	12	8	23	57
$n(E\|m_1, m_2)$	25	15	7	3
$n(E\|m_1, m_3)$	20	20	6	4
$n(E\|m_2, m_4)$	19	17	7	7
$n(E\|m_3, m_4)$	19	8	7	16
$n(E\|m_1, m_2, m_3, m_4)$	7	6	6	6

Each element in $\boldsymbol{S}$ and $\boldsymbol{n}$ pertain to an element in $\hat{\boldsymbol{\theta}}\{m_i\}$. For example, consider mode 1; the resulting vectors $\boldsymbol{S}$ and $\boldsymbol{n}$ are obtained as

$$
\begin{aligned}
\boldsymbol{S} &= [S(E|m_1), S(E|\{m_1, m_2\}), S(E|\{m_1, m_3\}), S(E|\{m_1, m_2, m_3, m_4\})] \\
&= [0.7, 0.5, 0.4, 0.28] \\
\boldsymbol{n} &= [n(m_1), n\{m_1, m_2\}, n\{m_1, m_3\}, n\{m_1, m_2, m_3, m_4\}] \\
&= [100, 50, 50, 25]
\end{aligned}
$$

Table 8.4: Support of modes containing m_1

	$E=[0,0]$	$E=[0,1]$	$E=[1,0]$	$E=[1,1]$
$S(E\|m_1)$	0.7	0.1	0.15	0.05
$S(E\|m_2)$	0.14	0.59	0.6	0.21
$S(E\|m_3)$	0.13	0.07	0.58	0.22
$S(E\|m_4)$	0.12	0.08	0.23	0.57
$S(E\|m_1,m_2)$	0.5	0.3	0.14	0.06
$S(E\|m_1,m_3)$	0.4	0.4	0.12	0.08
$n(E\|m_2,m_4)$	0.38	0.34	0.14	0.14
$n(E\|m_3,m_4)$	0.38	0.16	0.14	0.32
$S(E\|m_1,m_2,m_3,m_4)$	0.28	0.24	0.24	0.24

The function value, as well as the first and second derivative expressions are evaluated at $\hat{\Theta}$. For m_1, the parameter vector $\hat{\boldsymbol{\theta}}$ is given as:

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \left[1, \hat{\theta}\{\tfrac{m_1}{m_1,m_2}\}, \hat{\theta}\{\tfrac{m_1}{m_1,m_3}\}, \hat{\theta}\{\tfrac{m_1}{m_1,m_2,m_3,m_4}\}\right] \\ &= [1, 0.6/0.75, 0.6/0.75, 0.6/1.0] = [1, 0.8, 0.8, 0.6]\end{aligned}$$

Note that the terms $\boldsymbol{J}, \boldsymbol{H}$ only pertain to the variable terms in Θ, thus, in the case of m_1, $\theta\{\frac{m_1}{m_1}\} = 1$ is not included because it is a constant.

$$\boldsymbol{\theta} = \left[\theta\{\tfrac{m_1}{\{m_1,m_2\}}\}, \theta\{\tfrac{m_1}{\{m_1,m_3\}}\}, \theta\{\tfrac{m_1}{\{m_1,m_2,m_3,m_4\}}\}\right]$$

By taking derivatives according to the variable terms in Θ the Jacobian and Hessian can be obtained along with the reference value. Note that the derivative can be taken for other terms that are not variable, but their corresponding elements in $\boldsymbol{J}$ and $\boldsymbol{H}$ will be zero. For the purposes of notation compactness, we will only be taking the derivative with respect to $\boldsymbol{\theta}$, the variable elements in Θ for the mode in question. For example, when m_1 is selected, $\boldsymbol{J}$ and $\boldsymbol{H}$ take the following form:

$$\begin{aligned}\hat{p}(E|m_1) &= p(E|m_1, \hat{\Theta}) \\ \boldsymbol{J_i^{m_1}} &= \left[\begin{array}{ccc} \frac{\partial\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_1} & \frac{\partial\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_2} & \frac{\partial\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_3} \end{array}\right]_{\hat{\boldsymbol{\theta}}} \\ \boldsymbol{H_i^{m_1}} &= \left[\begin{array}{ccc} \frac{\partial^2\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_1^2} & \frac{\partial^2\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_1 \partial\ \boldsymbol{\theta}_2} & \frac{\partial^2\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_1 \partial\ \boldsymbol{\theta}_3} \\ \frac{\partial^2\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_2 \partial\ \boldsymbol{\theta}_1} & \frac{\partial^2\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_2^2} & \frac{\partial^2\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_2 \partial\ \boldsymbol{\theta}_3} \\ \frac{\partial^2\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_3 \partial\ \boldsymbol{\theta}_1} & \frac{\partial^2\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_3 \partial\ \boldsymbol{\theta}_2} & \frac{\partial^2\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_3^2} \end{array}\right]_{\hat{\boldsymbol{\theta}}}\end{aligned}$$

where the superscript m_1 pertains to mode 1. Using data from Tables 8.3 and 8.4, when $E=[0,0]$ is observed, the reference likelihood is:

$$\begin{aligned}p(E|m_1, \hat{\Theta}) &= \frac{\sum_k \boldsymbol{S}_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k}{\sum_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k} \\ &= \frac{0.7 \cdot 100 \cdot 1 + 0.5 \cdot 50 \cdot 0.8 + 0.4 \cdot 50 \cdot 0.8 + 0.28 \cdot 25 \cdot 0.6}{100 \cdot 1 + 50 \cdot 0.8 + 50 \cdot 0.8 + 25 \cdot 0.6} \\ &= \frac{70 + 20 + 16 + 4.2}{100 + 40 + 40 + 15} = 110.2/195 = 0.565\end{aligned}$$

In a similar manner, the Jacobian terms are calculated as

$$\frac{\partial^2\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_i^2} = \frac{\boldsymbol{n}_i \boldsymbol{S}_i}{\sum_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k} - \frac{\boldsymbol{n}_i\ \sum_k \boldsymbol{S}_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k}{\left(\sum_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k\right)^2}$$
$$= \frac{\boldsymbol{n}_i \boldsymbol{S}_i}{195} - \frac{\boldsymbol{n}_i\ 110.2}{195^2} = \frac{\boldsymbol{n}_i(195\ \boldsymbol{S}_i - 110.2)}{195^2}$$

Thus, the Jacobian can be expressed as

$$\boldsymbol{J^{m_1}} = \left[\ \frac{50(195(0.5)-110.2)}{195^2} \quad \frac{50(195(0.4)-110.2)}{195^2} \quad \frac{25(195(0.28)-110.2)}{195^2}\ \right]$$

Likewise, the Hessian terms can be expressed as

$$\left.\frac{\partial^2\ p(E|m_1,\Theta)}{\partial\ \boldsymbol{\theta}_i\ \partial\ \boldsymbol{\theta}_j}\right|_{\hat{\boldsymbol{\theta}}} = -\frac{\boldsymbol{n}_i \boldsymbol{S}_j + \boldsymbol{n}_j \boldsymbol{S}_i}{\left(\sum_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k\right)^2} + \frac{\boldsymbol{n}_i \boldsymbol{n}_j\ \sum_k \boldsymbol{S}_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k\{m\}}{\left(\sum_k \boldsymbol{n}_k \hat{\boldsymbol{\theta}}_k\right)^3}$$
$$= -\frac{\boldsymbol{n}_i \boldsymbol{S}_j + \boldsymbol{n}_j \boldsymbol{S}_i}{195^2} + \frac{\boldsymbol{n}_i \boldsymbol{n}_j 110.2}{195^3}$$
$$= \frac{\boldsymbol{n}_i \boldsymbol{n}_j \left(\frac{110.2}{195} - \frac{\boldsymbol{S}_j}{\boldsymbol{n}_j} - \frac{\boldsymbol{S}_i}{\boldsymbol{n}_i}\right)}{195^2}$$

so that the Hessian takes the following form

$$\boldsymbol{H^{m_1}} = \left[\begin{array}{ccc} \frac{50\cdot 50\left(\frac{110.2}{195}-\frac{0.5}{50}-\frac{0.5}{50}\right)}{195^2} & \frac{50\cdot 50\left(\frac{110.2}{195}-\frac{0.4}{50}-\frac{0.5}{50}\right)}{195^2} & \frac{50\cdot 25\left(\frac{110.2}{195}-\frac{0.28}{25}-\frac{0.5}{50}\right)}{195^2} \\ \frac{50\cdot 50\left(\frac{110.2}{195}-\frac{0.5}{50}-\frac{0.4}{50}\right)}{195^2} & \frac{50\cdot 50\left(\frac{110.2}{195}-\frac{0.4}{50}-\frac{0.4}{50}\right)}{195^2} & \frac{50\cdot 25\left(\frac{110.2}{195}-\frac{0.28}{25}-\frac{0.4}{50}\right)}{195^2} \\ \frac{25\cdot 50\left(\frac{110.2}{195}-\frac{0.5}{50}-\frac{0.28}{25}\right)}{195^2} & \frac{25\cdot 50\left(\frac{110.2}{195}-\frac{0.4}{50}-\frac{0.28}{25}\right)}{195^2} & \frac{25\cdot 25\left(\frac{110.2}{195}-\frac{0.28}{25}-\frac{0.28}{25}\right)}{195^2} \end{array}\right]$$

Note that these Jacobian and Hessian terms do not assume independence, and are calculated from joint probabilities of E_1, E_2; if independence is assumed, Jacobian and Hessian matrices have to be calculated for each independent piece of evidence.

8.3.7 Online Step 3: Perform combinations

For some modes, E_1 and E_2 are considered independent; if this is the case for mode M, second order terms exist for each piece of evidence. These terms must be combined before performing combination with the prior probabilities.

$$\hat{p}(E_1, E_2|M) = \hat{p}(E_1|M)\ \hat{p}(E_2|M)$$
$$\boldsymbol{J_L} = \hat{p}(E_1|M)\boldsymbol{J_2} + \hat{p}(E_2|M)\boldsymbol{J_1}$$
$$\boldsymbol{H_L} = \hat{p}(E_1|M)\boldsymbol{H_2} + \hat{p}(E_2|M)\boldsymbol{H_1} + \boldsymbol{J_2}^T\boldsymbol{J_1} + \boldsymbol{J_1}^T\boldsymbol{J_2}$$

where $\boldsymbol{J_L}$ and $\boldsymbol{H_L}$ are the Jacobian and Hessian of the overall likelihood. If E_1 and E_2 are considered dependent, then $\boldsymbol{J_L}$ and $\boldsymbol{H_L}$ are obtained directly from the previous step; no

combination is required. These likelihood terms are then combined with the prior terms $p(M)$, $\boldsymbol{J_P}$ and $\boldsymbol{H_P}$, in order to obtain the posterior terms $p(M|E_1, E_2)$, $\boldsymbol{J_F}$ and $\boldsymbol{H_F}$

$$\begin{aligned} K &= \sum_k \hat{p}(m_k)\hat{p}(E_1, E_2|M) \\ \hat{p}(M|E_1, E_2) &= \frac{1}{K}\hat{p}(M)\hat{p}(E_1, E_2|M) \\ \boldsymbol{J_F} &= \frac{1}{K}\left[\hat{p}(M)\boldsymbol{J_L} + \hat{p}(E_1, E_2|M)\boldsymbol{J_P}\right] \\ \boldsymbol{H_F} &= \frac{1}{K}\left[\hat{p}(M)\boldsymbol{H_L} + \hat{p}(E_1, E_2|M)\boldsymbol{H_P} + \boldsymbol{J_L}^T\boldsymbol{J_P} + \boldsymbol{J_P}^T\boldsymbol{J_L}\right] \end{aligned}$$

Keep in mind that priors focus on the frequency of unambiguous modes; thus $\boldsymbol{J_F}$ and $\boldsymbol{H_F}$ tend to be zero matrices.

In the case of our example, $\hat{p}(M)$ is given in Table 8.2; the likelihood reference $\hat{p}(E_1, E_2|m_1)$ was already given to be 0.565; by calculating the reference likelihoods for other modes, we get the likelihood vector $\hat{p}(E_1, E_2|M) = [0.565, 0.198, 0.184, 0.168]$. From this, the normalization constant is

$$\begin{aligned} K &= \sum_k \hat{p}(m_1)\hat{p}(E_1, E_2|m_1) \\ &= 0.565 \cdot 0.6 + 0.198 \cdot 0.15 + 0.184 \cdot 0.15 + 0.168 \cdot 0.1 = 0.413 \end{aligned}$$

When keeping in mind previous Jacobian and Hessian terms $\boldsymbol{J^{m_1}}, \boldsymbol{H^{m_1}}$, and that Jacobian and Hessian terms associated with prior probabilities are zero matrices, the posterior probability along with the Jacobian and Hessian matrices are obtained as

$$\begin{aligned} \hat{p}(m_1|E_1, E_2) &= \frac{1}{0.413}(0.6 \cdot 0.565) = 0.821 \\ \boldsymbol{J_F^{m_1}} &= \frac{1}{0.413}\left(0.6\boldsymbol{J^{m_1}} + 0.565[\mathbf{0}]\right) = 1.45\ \boldsymbol{J^{m_1}} \\ &= \begin{bmatrix} -0.0228 & -0.0579 & -0.0500 \end{bmatrix} \\ \boldsymbol{H_F^{m_1}} &= \frac{1}{0.413}\left(0.6\boldsymbol{H^{m_1}} + 0.565[\mathbf{0}] + (\boldsymbol{J^{m_1}})^T\mathbf{0} + \mathbf{0}^T J^{m_1}\right) = 1.45\ \boldsymbol{H^{m_1}} \\ &= \begin{bmatrix} 0.0490 & 0.0492 & 0.0244 \\ 0.0492 & 0.0493 & 0.0245 \\ 0.0244 & 0.0245 & 0.0122 \end{bmatrix} \end{aligned}$$

8.3.8 Online Step 4: Make a diagnosis

Diagnosis using the point estimate $\hat{p}(M|E_1, E_2)$

A diagnosis can be made using the reference posterior probabilities $\hat{p}(M|E_1, E_2)$. This is the simplest method available and does not require the calculation of $\boldsymbol{J_F}$ and $\boldsymbol{H_F}$. The diagnosis is made by selecting the mode with largest corresponding value of $\hat{p}(M|E_1, E_2)$.

In our example, the set of posterior probabilities was found to be

$$\hat{p}(M|E_1,E_2) = \frac{1}{0.413}\begin{bmatrix} 0.565\times 0.6 \\ 0.198\times 0.15 \\ 0.184\times 0.15 \\ 0.168\times 0.1 \end{bmatrix} = \begin{bmatrix} 0.8207 \\ 0.0719 \\ 0.0668 \\ 0.0406 \end{bmatrix}$$

From this example, one would diagnose mode 1.

Diagnosis using the expected value of $p(M|E_1,E_2)$

Instead of assuming a point value for $p(M|E_1,E_2)$ using $\hat{\Theta}$, we could assume a distribution over the possible values of Θ in order to calculate the expected value of $p(M|E_1,E_2)$. Consider mode 1 with its corresponding values $\boldsymbol{J_F}$ and $\boldsymbol{H_F}$ along with the corresponding parameter vector $\boldsymbol{\theta}$ from Θ that can be varied

$$\begin{aligned}\boldsymbol{\theta} &= \left[\theta\{\tfrac{m_1}{\{m_1,m_2\}}\},\theta\{\tfrac{m_1}{\{m_1,m_3\}}\},\theta\{\tfrac{m_1}{\{m_1,m_2,m_3,m_4\}}\}\right]^T \\ &= [\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\boldsymbol{\theta}_3]^T\end{aligned}$$

Note that because $\theta\{\frac{m_1}{m_1}\} = 1$ is not variable but constant, it is now excluded from $\boldsymbol{\theta}$. The vector of expected values and expected squared values are given as

$$\begin{aligned}E[\boldsymbol{\theta}] &= [E(\boldsymbol{\theta}_1),E(\boldsymbol{\theta}_2),E(\boldsymbol{\theta}_3)]^T \\ E[\boldsymbol{\theta}^2] &= \left[E(\boldsymbol{\theta}_1^2),E(\boldsymbol{\theta}_2^2),E(\boldsymbol{\theta}_3^2)\right]^T\end{aligned}$$

These expected values are calculated using Eqn (8.6) and (8.7)

$$\begin{aligned}E[\theta\{\tfrac{m_1}{\boldsymbol{m}_k}\}] &= \frac{\alpha(\frac{m_1}{\boldsymbol{m}_k})}{\sum\limits_{m_i\subset\boldsymbol{m}_k}\alpha(\frac{m_1}{\boldsymbol{m}_k})} \\ E[\theta^2\{\tfrac{m_1}{\boldsymbol{m}_k}\}] &= \frac{\alpha(\frac{m_1}{\boldsymbol{m}_k})+\alpha^2(\frac{m_1}{\boldsymbol{m}_k})}{\left[\sum\limits_{m_i\subset\boldsymbol{m}_k}\alpha(\frac{m_1}{\boldsymbol{m}_k})\right]+\left[\sum\limits_{m_i\subset\boldsymbol{m}_k}\alpha(\frac{m_1}{\boldsymbol{m}_k})\right]^2}\end{aligned}$$

where $\alpha(\frac{m_i}{\boldsymbol{m}_k})$ is calculated using

$$\alpha(\tfrac{m_i}{\boldsymbol{m}_k}) = \hat{\theta}\{\tfrac{m_i}{\boldsymbol{m}_k}\}n(\boldsymbol{m}_k)$$

As an example, let us consider the prior probabilities in Table 8.2 and the mode frequencies in Table 8.3, then the resulting α parameters would be

$$\begin{aligned}\alpha(\tfrac{m_1}{m_1,m_2}) &= \hat{\theta}\{\tfrac{m_1}{m_1,m_2}\}n(m_1,m_2) \\ &= 0.8\times 50 = 40\end{aligned}$$

The value for α conveys a degree of certainty on $\hat{\theta}\{\frac{m_1}{m_1,m_2}\}$ (where $\alpha\to\infty$ indicates complete certainty). If one wishes to reduce certainty on α it is possible to scale the values for α by

a value less than 1 (scaling by zero conveys complete uncertainty). Using these values for α, the expectation $E(\boldsymbol{\theta}_1)$ is calculated as

$$\begin{aligned} E(\boldsymbol{\theta}_1) &= E[\theta\{\tfrac{m_1}{m_1,m_2}\}] = \frac{\alpha(\frac{m_1}{m_1,m_2})}{\alpha(\frac{m_1}{m_1,m_2}) + \alpha(\frac{m_2}{m_1,m_2})} \\ &= \frac{40}{40+10} = 0.8 \\ E(\boldsymbol{\theta}_1^2) &= E[\theta^2\{\tfrac{m_1}{m_1,m_2}\}] = \frac{\alpha(\frac{m_1}{m_1,m_2}) + \alpha^2(\frac{m_1}{m_1,m_2})}{[\alpha(\frac{m_1}{m_1,m_2}) + \alpha(\frac{m_1}{m_1,m_2})] + [\alpha(\frac{m_1}{m_1,m_2}) + \alpha(\frac{m_1}{m_1,m_2})]^2} \\ &= \frac{40+40^2}{50+50^2} = 0.643 \end{aligned}$$

When this techniques applied to m_1 using the other ambiguous modes $\{m_1, m_3\}, \{m_1, m_2, m_3, m_4\}$ the elements of $E(\boldsymbol{\theta})$ and $E(\boldsymbol{\theta}^2)$ are given as follows:

$$\begin{aligned} E(\boldsymbol{\theta}) &= [0.8, 0.8, 0.6]^T \\ E(\boldsymbol{\theta}^2) &= [0.643, 0.643, 0.369]^T \end{aligned}$$

The next step is to obtain all the terms in the expression for the second-order expectation

$$E[p(M|E)] = \boldsymbol{C} + \boldsymbol{J}^* E[\boldsymbol{\theta}] + \frac{1}{2}\left[E[\boldsymbol{\theta}]^T \boldsymbol{H}_{OD} E[\boldsymbol{\theta}] + E[\boldsymbol{\theta}^2]^T \boldsymbol{H}_D\right]$$

The first terms $\boldsymbol{C}$ and $\boldsymbol{J}^*$ are obtained using the reference parameter values $\hat{\boldsymbol{\theta}}$. Note that because the non-variable terms in $\hat{\boldsymbol{\theta}}$ have zero Jacobian and Hessian elements, they are omitted.

$$\begin{aligned} \boldsymbol{C} &= \hat{p}(M|E) - \boldsymbol{J}\hat{\boldsymbol{\theta}} + \frac{1}{2}\hat{\boldsymbol{\theta}}^T \boldsymbol{H} \hat{\boldsymbol{\theta}} \\ &= 0.821 - \begin{bmatrix} -0.0228 \\ -0.0579 \\ -0.0500 \end{bmatrix}^T \begin{bmatrix} 0.8 \\ 0.8 \\ 0.6 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0.8 \\ 0.8 \\ 0.6 \end{bmatrix}^T \begin{bmatrix} 0.0490 & 0.0492 & 0.0244 \\ 0.0492 & 0.0493 & 0.0245 \\ 0.0244 & 0.0245 & 0.0122 \end{bmatrix} \begin{bmatrix} 0.8 \\ 0.8 \\ 0.6 \end{bmatrix} \\ &= 0.725 \\ \boldsymbol{J}^* &= \boldsymbol{J} - \hat{\boldsymbol{\theta}}^T \boldsymbol{H} \\ &= \begin{bmatrix} -0.0228 \\ -0.0579 \\ -0.0500 \end{bmatrix}^T - \begin{bmatrix} 0.8 \\ 0.8 \\ 0.6 \end{bmatrix}^T \begin{bmatrix} 0.0490 & 0.0492 & 0.0244 \\ 0.0492 & 0.0493 & 0.0245 \\ 0.0244 & 0.0245 & 0.0122 \end{bmatrix} \\ &= \begin{bmatrix} -0.0910 - 0.1263 - 0.0840 \end{bmatrix} \end{aligned}$$

Next, $\boldsymbol{H}_{OD}$ is obtained by subtracting the main diagonal of $\boldsymbol{H}$ from H, meanwhile $\boldsymbol{H}_D$ is a vector of the main diagonal of $\boldsymbol{H}$ that was removed from $\boldsymbol{H}_{OD}$. For our example, $\boldsymbol{H}_{OD}$ and $\boldsymbol{H}_D$ are given as

$$\begin{aligned} \boldsymbol{H}_{\boldsymbol{OD}}^{\boldsymbol{1}} &= \begin{bmatrix} 0 & 0.0492 & 0.0244 \\ 0.0492 & 0 & 0.0245 \\ 0.0244 & 0.0245 & 0 \end{bmatrix} \\ \boldsymbol{H}_{\boldsymbol{D}}^{\boldsymbol{1}} &= \begin{bmatrix} 0.0490 & 0.0493 & 0.0122 \end{bmatrix} \end{aligned}$$

With all of the expectation vectors and matrix terms defined, the expected value $E[p(M|E)]$ can be calculated according to Eqn 8.5.

$$\begin{aligned} E[p(M|E)] &= \boldsymbol{C} + \boldsymbol{J}^* E[\boldsymbol{\theta}] + \frac{1}{2}\left[E[\boldsymbol{\theta}]^T \boldsymbol{H}_{OD} E[\boldsymbol{\theta}] + E[\boldsymbol{\theta}^2]^T \boldsymbol{H}_D\right] \\ &= 0.821 \end{aligned}$$

which is almost identical to the posterior probability of mode (1) obtained from the point estimate.

Diagnosis using the probability ranges of $p(M|E_1, E_2)$

While the simple method and the expectation method yield point estimates of the probability, the second-order approximation can also be used to obtain probability ranges. The probability is a function of Θ given by

$$\begin{aligned} \Delta\Theta &= \Theta - \hat{\Theta} \\ p(M|E,\Theta) &= \hat{p}(M|E) + \boldsymbol{J}_{\boldsymbol{F}}^{\boldsymbol{m}} \Delta\Theta + \frac{1}{2}\Delta\Theta^T \boldsymbol{H}_{\boldsymbol{F}}^{\boldsymbol{m}} \Delta\Theta \end{aligned}$$

The belief or lower bound probability is obtained by minimizing $p(M|E,\Theta)$ subject to the constraint $\mathbf{0} \leq \Theta \leq \mathbf{1}$ while the plausibility or upper bound probability is obtained by maximizing $p(M|E,\Theta)$ subject to the same constraints. Because this is a constrained quadratic expression, the maximization and minimization problems can be solved using standard quadratic programming techniques.

While the actual diagnosis can be obtained using either the point estimate or expected value of $p(M|E_1, E_2)$, the belief and plausibility can give additional information about the uncertainty of the diagnosis. Consider a hypothetical result in Figure 8.2 where point estimate is given by a dotted line, and the uncertainty regions are given in a lighter shades of gray. In this example, the uncertainty region for mode 2 and mode 3 overlap. Thus while the system is most likely operating at mode 2, it would be unwise to rule out mode 3, as under certain circumstances, mode 3 could be more probable than mode 2.

8.4 Simulated Case

The proposed second-order method was tested on the simulated Tenessee Eastman problem. In this simulation, data was masked as ambiguous based on its resemblance toward other modes. In the masking process, distributions were estimated for each mode and a likelihood ratio threshold was set, so that if another mode was likely enough, the data point was classified as ambiguous. For example, consider the data from mode 1. If there is a data point d_i (consisting of evidence) where the likelihood ratio R between mode k and mode 1 was large enough

$$R = \frac{p(d_i|m_k)}{p(d_i|m_1)} > \text{Threshold}$$

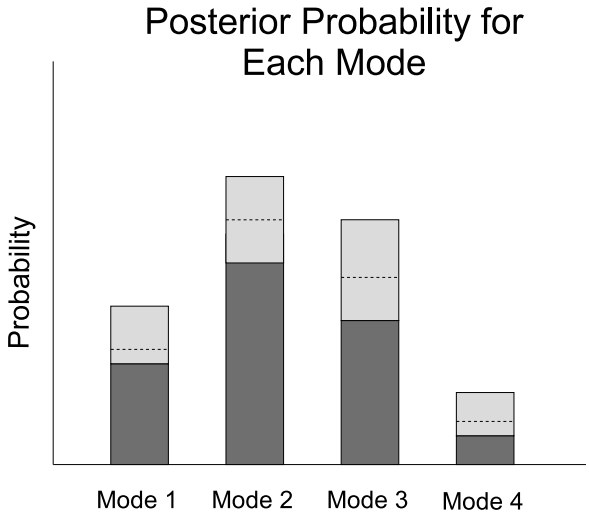
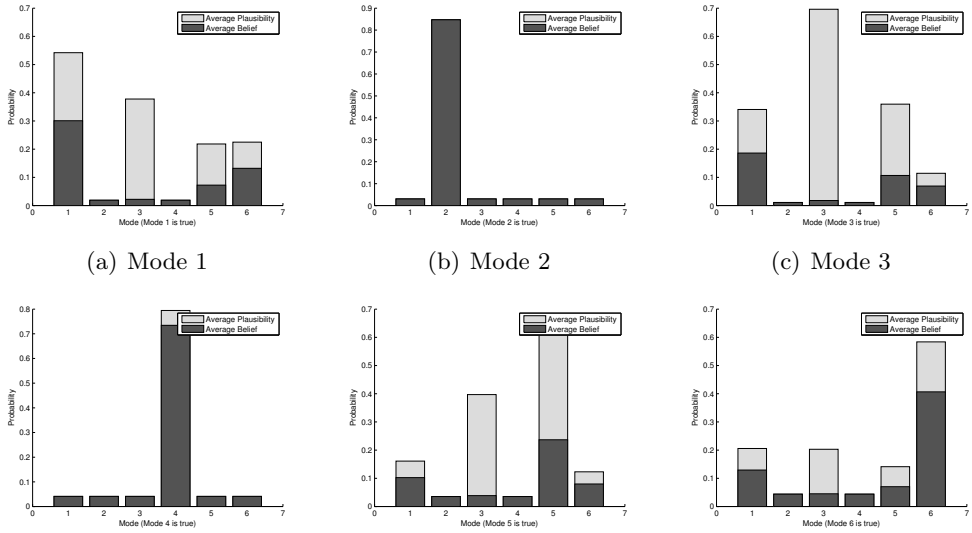


Figure 8.2: Typical control loop

then the mode m_k was added, rendering the mode associated with d_i ambiguous. By manipulating the threshold, certain amounts of data can be classified as ambiguous. For this simulation case, data was abundant, and the monitors showed fairly clear results. In order to demonstrate the effectiveness of the second-order method, 50% of data was deliberately removed from the history, and noise was added.

First, we took a look at the average probability boundaries given for the system. Here, second order posterior terms were calculated and boundaries were calculated via quadratic programming. The first set of figures (Figure 8.3) had an ambiguity threshold set so that 30 % of the data had modes with ambiguity. In the second set of figures (Figure 8.4) each had 70 % of the data belonging to ambiguous modes. From these figures, one can see that modes 2 and 4 were the easiest to diagnose, with the true probability being notably higher, and with small probability boundaries. Modes 3 and 5 were the most difficult, with much larger probability boundaries and a tendency toward mutual confusion. When the amount of ambiguous mode data was increased from 30% to 70%, probability ranges grew, primarily from modes 1 and 6; however, there was a slight increase in probability boundaries from modes 2 and 4. Probability boundaries for modes 3 and 5 were already quite large with 30% ambiguous mode data; increasing the amount of ambiguous mode data from 30% to 70% had little effect.

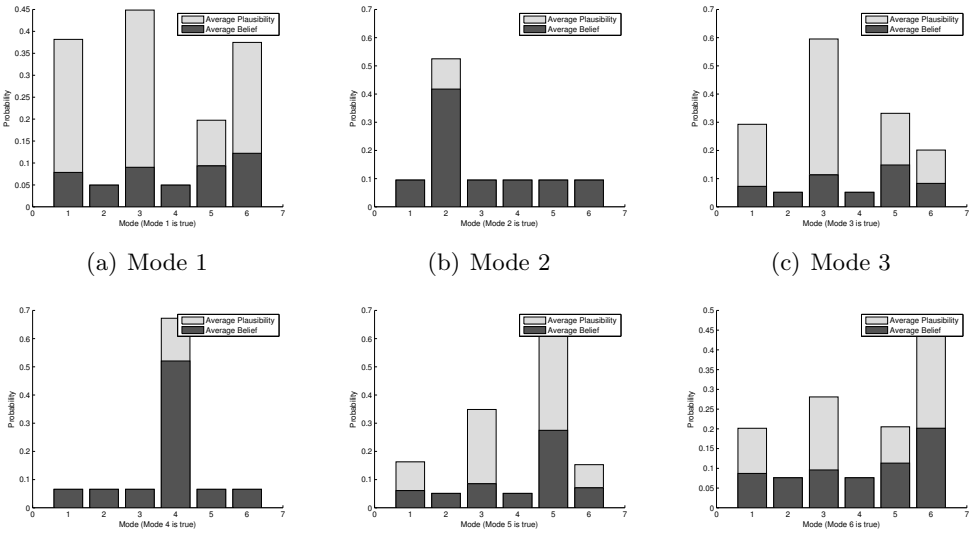
After evaluating the probability boundaries, we proceed to evaluating overall diagnosis performance. For the sake of simplicity diagnosis was based on the point estimate method (from the informed transformation $\hat{p}(M|E) = p(M|E, \hat{\Theta})$). In Figure 8.5, mode diagnosis performance was assessed. The performance metric used is percent of misdiagnosed modes. Three different methods were compared:



(a) Mode 1 (b) Mode 2 (c) Mode 3

(d) Mode 4 (e) Mode 5 (f) Mode 6

Figure 8.3: Probablility bounds at 30 % ambiguity



(a) Mode 1 (b) Mode 2 (c) Mode 3

(d) Mode 4 (e) Mode 5 (f) Mode 6

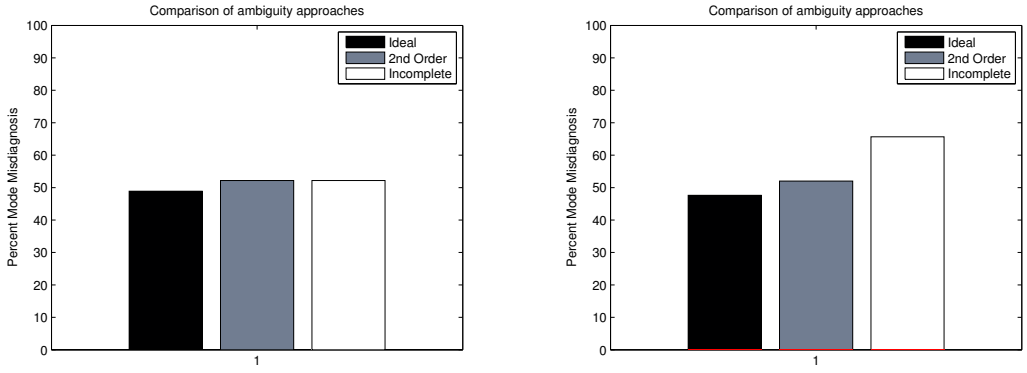
Figure 8.4: Probablility bounds at 70 % ambiguity

1. **Ideal Case:** This refers to the case where ambiguity is not present. The original Bayesian method is performed on data before masking techniques were applied.

2. **2nd Order Method:** This referes to the case where the second-order Bayesian method is performed on data containing ambiguous modes.

3. **Incomplete Bayesian Method:** This refers to the case where ambiguous modes were present in the data, but they were ignored so that the original Bayesian method can be performed on the remaining data set.

The second-order method was compared to the *incomplete Bayesian method*, where masked data was simply discarded, and the *ideal case*, where masked data was known. In Figure 8.5(a), the three methods performed very similarly thus ignoring the 30 % of the data that was ambiguous did not significantly affect the result. However, when the amount of ambiguous data increased to 70 %, the incomplete Bayesian method ignores a significant amount of data. Thus when there was more ambiguous data, the performance improvement brought on by the second-order method was significant.



(a) 30 % Ambiguous Mode Data

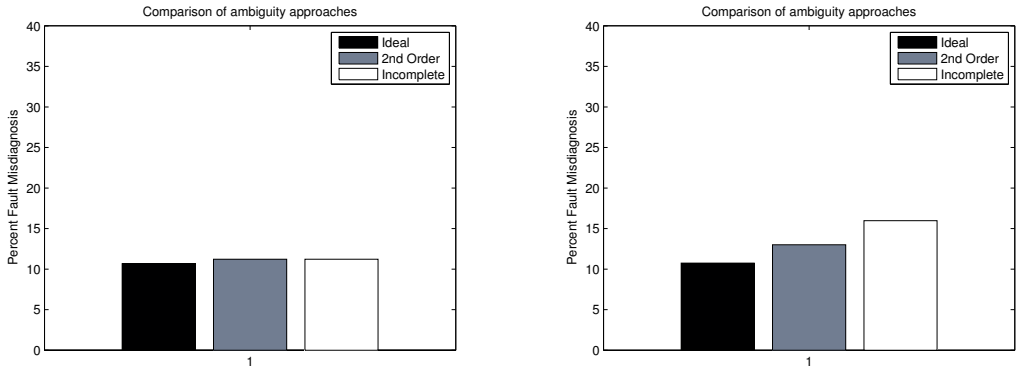
(b) 70 % Ambiguous Mode Data

Figure 8.5: TE mode diagnosis error

In addition to diagnosing modes, the state of each individual component was diagnosed. For this system, the component states were:

1. A/C feed ratio step change (stream 4)
2. B composition step change (stream 4)
3. C header pressure loss
4. A,B,C, Feed Composition step change (stream 4)
5. D feed temperature (stream 2)
6. Reactor cooling water inlet temperature
7. Sticky reactor cooling water valve

In many cases, if a mode is incorrectly diagnosed, the correct mode is similar, usually differing by one or two components. Consequently, component diagnosis tends to exhibit better performance with fewer false results. Unsurprisingly, when diagnosing components, the second-order method showed a performance improvement that was similar to the improvement shown when diagnosing modes. Results are shown in Figure 8.6.



(a) 30 % Ambiguous Mode Data

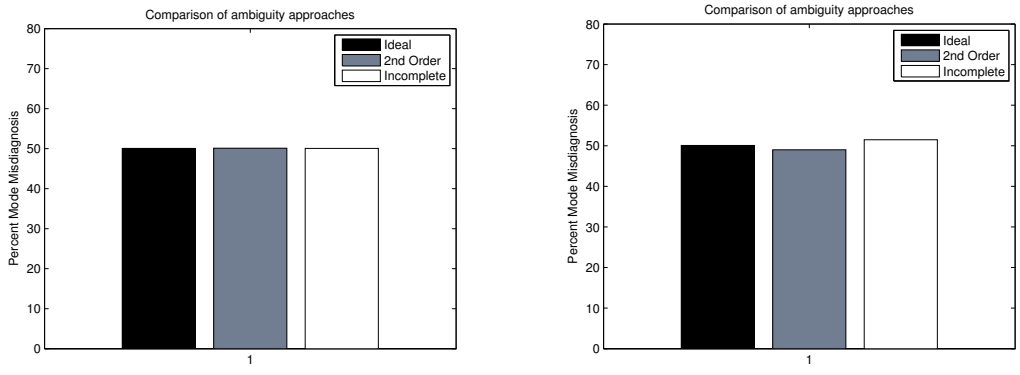
(b) 70 % Ambiguous Mode Data

Figure 8.6: TE component diagnosis error

8.5 Bench Scale Case

This method was also applied to the lab-scale hybrid tank system where tank leaks and sensor bias were to be detected. Again, data was masked based on its proximity to other modes and diagnosis was based on the point estimate given by the informed transformation $\hat{p}(M|E) = p(M|E, \hat{\Theta})$.

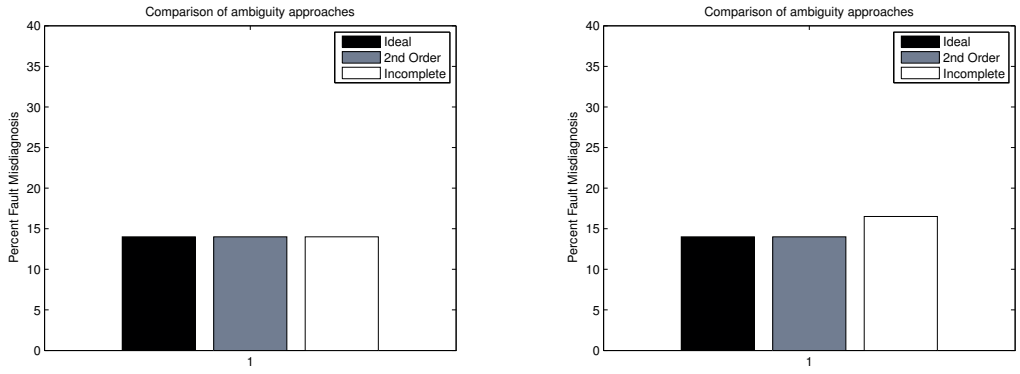
Mode diagnosis results are available in Figure 8.7. When diagnosing modes, performance seemed to be relatively unchanged even when 70% of the data was missing; this robustness toward missing data was again likely due to the abundance of data. In contrast, Figure 8.8 shows results when diagnosing components, where it can be seen that the second-order approach shows a slightly more significant improvement. While the frequency of diagnosing the correct mode may not have significantly improved when the second order method was applied, it appears that the incorrectly diagnosed modes bear more resemblance to the true mode.



(a) 30 % Ambiguous Mode Data

(b) 70 % Ambiguous Mode Data

Figure 8.7: Hybrid tank system mode diagnosis error



(a) 30 % Ambiguous Mode Data

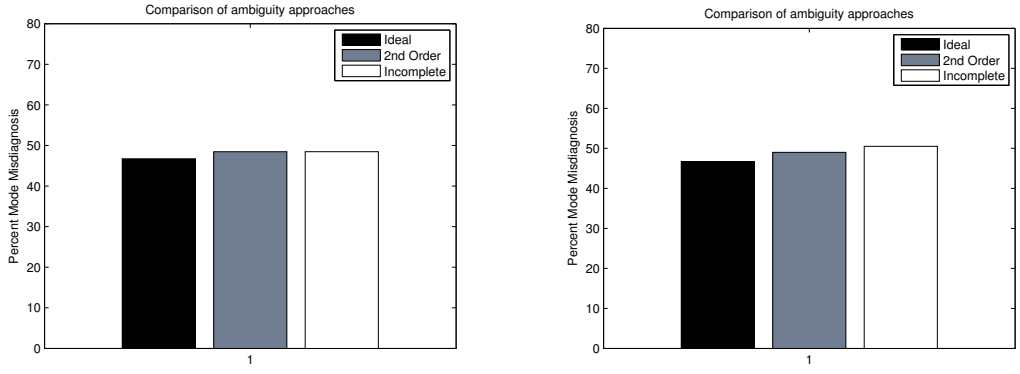
(b) 70 % Ambiguous Mode Data

Figure 8.8: Hybrid tank system component diagnosis error

8.6 Industrial Scale Case

Finally, this method was tested on industrial data, where the on and off conditions were detected for each subsystem. Data was masked by taking into account proximity with other modes and again, diagnosis results were based on the point estimate of the posterior probability $\hat{p}(M|E) = p(M|E, \hat{\Theta})$.

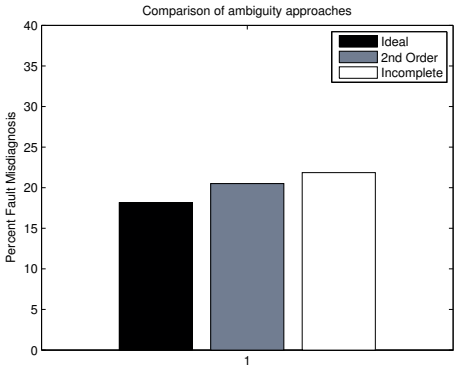
Mode diagnosis results are shown in Figure 8.9. In a similar manner as the experimental system, diagnosis results are fairly uniform for both the cases with 30 % ambiguous mode data and 70 % ambiguous mode data. When diagnosing components however (Figure 8.10), the improvements for the second-order method appear to be more pronounced; thus, while a similar number of modes may still be incorrectly diagnosed, it is evident that the incorrect modes tend to resemble the correct mode more closely.



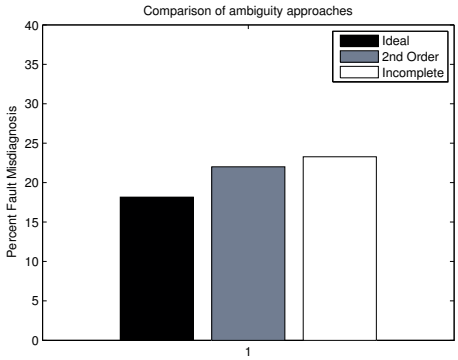
(a) 30 % Ambiguous Mode Data

(b) 70 % Ambiguous Mode Data

Figure 8.9: Industrial system mode diagnosis error



(a) 30 % Ambiguous Mode Data



(b) 70 % Ambiguous Mode Data

Figure 8.10: Industrial system component diagnosis error

Chapter 9

Accounting for ambiguous modes in historical data: A Dempster-Shafer approach

9.1 Introduction

Dempster-Shafer theory (DST) has long been considered as a more general alternative to Bayesian combination; DST directly accounts for ambiguity and has had a wide variety of adaptation in the literature. However, Dempster-Shafer theory does not lend itself well to making inference from history-based likelihoods containing ambiguous modes; becuase of this, DST needs to be generalized in order better fit this task.

As in Bayesian combination, the goal is to combine information from independent sources of evidence (including prior probabilities). The application of Generalized DST (or GDST) takes the same form as the second-order method discussed in Chapter 8. This chapter gives details on how to

1. Set up a method for calculating likelihoods given θ parameters $p(E|M, \mathbf{\Theta})$)
2. Calculate the Generalized Basic Belief Assignment (GBBA) for both priors and likelihoods
3. Combine the GBBAs to obtain a final diagnosis result
4. Group monitors together into independent groups (optional)

9.2 Algorithm

9.2.1 Parametrized Likelihoods

Likelihoods are parametrized in the same way as in Chapter 8

$$p(E|M,\Theta) = \frac{\sum\limits_{\boldsymbol{m}_k \supseteq m} \theta\{\frac{m}{\boldsymbol{m}_k}\} S(E|\boldsymbol{m}_k) n(\boldsymbol{m}_k)}{\sum\limits_{\boldsymbol{m}_k \supseteq m} \theta\{\frac{m}{\boldsymbol{m}_k}\} n(\boldsymbol{m}_k)} \tag{9.1}$$

where $n(\boldsymbol{m}_k)$ is the number of times $\boldsymbol{m}_k$ appears in the history, and $S(E|\boldsymbol{m}_k)$ is the support function for evidence E given $\boldsymbol{m}_k$, which, for discrete data is calculated as

$$S(E|\boldsymbol{m}_k) = \frac{n(E|\boldsymbol{m}_k)}{n(\boldsymbol{m}_k)}$$

In this chapter, Eqn (9.1) is used as a basis for calculating Generalized BBAs (or GBBAs).

9.2.2 Basic Belief Assignments

Traditional Basic Belief Assignments

The Basic Belief Assignment (BBA) is a function with respect to the mode and is denoted as $S(\boldsymbol{m}_k)$; it is similar to probability except that it can yield support to ambiguous modes as well (the boldface $\boldsymbol{m}$ is used to represent potential ambiguity). The probability $p(M|E)$ is calculated using $S(\boldsymbol{m}|E)$ according to

$$p(M|E,\Theta) = \sum_{\boldsymbol{m}_k \supseteq m} \theta\{\tfrac{m}{\boldsymbol{m}_k}\} S(\boldsymbol{m}_k|E) \tag{9.2}$$

where the support function (or BBA) $S(\boldsymbol{m}|E)$ is given as

$$S(\boldsymbol{m}|E) = \frac{n(\boldsymbol{m},E)}{n(E)} \tag{9.3}$$

Here, $n(\boldsymbol{m},E)$ is the number of times the mode $\boldsymbol{m}$ and evidence E jointly occur, while $n(E)$ is the total number of times evidence E occurs. In addition to probability, Dempster-Shafer theory also concerns itself with belief (the lower-bound probability) and plausibility (the upper-bound probability). Because $p(M|E,\Theta)$ is linear with respect to θ and the coefficients $S(\boldsymbol{m})$ on θ are positive, $p(M|E,\Theta)$ is maximized by setting θ to 1 whenever possible and is minimized by setting θ to 0 whenever possible.

$$Bel(M|E,\Theta) = \min_{\Theta}\left[p(M|E,\Theta)\right] = \sum_{\boldsymbol{m}_k \subseteq m} S(\boldsymbol{m}_k|E)$$
$$Pl(M|E,\Theta) = \max_{\Theta}\left[p(M|E,\Theta)\right] = \sum_{\boldsymbol{m}_k \supseteq m} S(\boldsymbol{m}_k|E)$$

Unfortunately, the form of Eqn (9.2) is too restrictive to adequately represent the base problem in Eqn (9.1). The two most detrimental restrictions of Eqn (9.2) lie in the terms

$S(\boldsymbol{m})$ (which function as coefficients on Θ). The first restriction is that terms in $S(\boldsymbol{m})$ are positive, and the second restriction is that terms must be identical for all unambiguous modes contained in $\boldsymbol{m}$. The reasons for relaxing these restrictions have been discussed in Chapter 5, but they mainly boil down to the fact that Eqn (9.1) is non-linear with respect to Θ. The task at hand is to obtain a generalized expression of the BBA that better represents the problem presented in Eqn (9.1). This can be done by allowing the BBA to have negative support, and to have different amounts of support for different modes in $\boldsymbol{m}$.

Generalized Basic Belief Assignments

The generalized BBA (or GBBA) denoted as $\boldsymbol{G}$ is a matrix that can approximate the expression of $p(E|M,\Theta)$ in Eqn (9.2)

$$p(E|M,\boldsymbol{\Theta}) = \boldsymbol{G}[:,m]^T\boldsymbol{\Theta}[:,m] \tag{9.4}$$

where $\boldsymbol{\Theta}$ is the matrix form of Θ and $\boldsymbol{\Theta}[:,m]$ is the column of $\boldsymbol{\Theta}$ that pertains to the specific mode m ($\boldsymbol{G}[:,m]^T$ is likewise the column of $\boldsymbol{G}$ pertaining to m). The expression in Eqn (9.4) is a first-order approximation of Eqn (9.1)

As an example for the structure of $\boldsymbol{G}$, let us consider a three mode system m_1, m_2, m_3 with ambiguous modes $\{m_1,m_2\}$, $\{m_2,m_3\}$, $\{m_1,m_3\}$. The structure of $\boldsymbol{G}$ and $\boldsymbol{\Theta}$ are given as

$$\boldsymbol{G} = \left[\begin{array}{c|ccc} & m_1 & m_2 & m_3 \\ \hline m_1 & G\{\frac{m_1}{m_1}\} & 0 & 0 \\ m_2 & 0 & G\{\frac{m_2}{m_2}\} & 0 \\ m_3 & 0 & 0 & G\{\frac{m_3}{m_3}\} \\ \{m_1,m_2\} & G\{\frac{m_1}{m_1,m_2}\} & G\{\frac{m_2}{m_1,m_2}\} & 0 \\ \{m_1,m_3\} & G\{\frac{m_1}{m_1,m_3}\} & 0 & G\{\frac{m_3}{m_1,m_3}\} \\ \{m_2,m_3\} & 0 & G\{\frac{m_2}{m_2,m_3}\} & G\{\frac{m_3}{m_2,m_3}\} \end{array}\right]$$

$$\boldsymbol{\Theta} = \left[\begin{array}{c|ccc} & m_1 & m_2 & m_3 \\ \hline m_1 & 1 & 0 & 0 \\ m_2 & 0 & 1 & 0 \\ m_3 & 0 & 0 & 1 \\ \{m_1,m_2\} & \theta\{\frac{m_1}{m_1,m_2}\} & \theta\{\frac{m_2}{m_1,m_2}\} & 0 \\ \{m_1,m_3\} & \theta\{\frac{m_1}{m_1,m_3}\} & 0 & \theta\{\frac{m_3}{m_1,m_3}\} \\ \{m_2,m_3\} & 0 & \theta\{\frac{m_2}{m_2,m_3}\} & \theta\{\frac{m_3}{m_2,m_3}\} \end{array}\right]$$

Any value of $\boldsymbol{\Theta}$ that is not set to 0 or 1 is considered to be unknown or *flexible*. The approximation for $p(E|M,\boldsymbol{\Theta})$ is then given as

$$\begin{aligned} p(E|M_1,\boldsymbol{\Theta}) &= \boldsymbol{G}[:,m_1]^T\boldsymbol{\Theta}[:,m_1] \\ &= G\{\tfrac{m_1}{m_1}\}\theta\{\tfrac{m_1}{m_1}\} + G\{\tfrac{m_1}{m_1,m_2}\}\theta\{\tfrac{m_1}{m_1,m_2}\} + G\{\tfrac{m_1}{m_1,m_3}\}G\{\tfrac{m_1}{m_1,m_3}\} \end{aligned}$$

The individual terms of $\boldsymbol{G}$ can be calculated according to the following heuristic

$$\boldsymbol{G}[k,i] = \begin{cases} 0 & m_i \cap \boldsymbol{m}_k = \emptyset \\ \tilde{p}(E|m_i) & m_i = \boldsymbol{m}_k \\ \frac{\partial p(E|m_i)}{\partial \boldsymbol{\theta}[k,i]} & m_i \subset \boldsymbol{m}_k \end{cases} \tag{9.5}$$

where

$$\tilde{p}(E|M) = p(E|M,\hat{\boldsymbol{\Theta}}) - \sum_{\boldsymbol{m}_k \supset M} \hat{\theta}\{\tfrac{M}{\boldsymbol{m}_k}\} \left. \frac{\partial\, p(E|M,\boldsymbol{\Theta})}{\partial\, \theta\{\frac{M}{\boldsymbol{m}_k}\}} \right|_{\hat{\boldsymbol{\Theta}}}$$

$$\frac{\partial p(E|M)}{\partial \boldsymbol{\Theta}[k,i]} = \left. \frac{\partial\, p(E|M,\boldsymbol{\Theta})}{\partial\, \theta\{\frac{M}{\boldsymbol{m}_k}\}} \right|_{\hat{\boldsymbol{\Theta}}}$$

Here, $\hat{\boldsymbol{\Theta}}$ is the reference value of $\boldsymbol{\Theta}$ which, for the generalized Dempster-Shafer method is set to the *inclusive value* of $\boldsymbol{\Theta}$.

$$\hat{\boldsymbol{\Theta}} = \boldsymbol{\Theta}^*$$

where

- $\boldsymbol{\Theta}^*$ is the *inclusive value*, which is defined by setting all flexible values to 1
- $\boldsymbol{\Theta}_*$ is the *exclusive value*, which is defined by setting all flexible values to 0

When $\boldsymbol{G}$ is defined in this manner, Eqn (9.4) is a first-order Taylor Series approximation of $p(E|m_1,\boldsymbol{\Theta})$.

Belief and plausibility can also be calculated from Eqn (9.4) by minimizing and maximizing $p(E|m_1,\boldsymbol{\Theta})$.

$$Bel(E|M) = \min_{\boldsymbol{\Theta}} \boldsymbol{G}[:,m]^T \boldsymbol{\Theta}[:,m]$$
$$Pl(E|M) = \max_{\boldsymbol{\Theta}} \boldsymbol{G}[:,m]^T \boldsymbol{\Theta}[:,m]$$

where $\boldsymbol{\Theta}$ is the set of variable elements in $\boldsymbol{\Theta}[:,m]$ (i.e. the elements not automatically set to 1 or 0 by logic). Again, the variables values in $\boldsymbol{\Theta}$ are constrained to be between 0 and 1. However, one should note that the elements in $\boldsymbol{G}$ are no longer positive. Consequently, when calculating belief and plausibility, the minimum is no longer obtained by setting all flexible values in $\boldsymbol{\Theta}$ to zero; likewise, the maximum is no longer maximized by setting all flexible values in $\boldsymbol{\Theta}$ to one. Instead, the belief and plausibility should be obtained using linear programming methods.

9.2.3 The Generalized Dempster's Rule of combination

In the same way as Bayesian methods, the Generalized Dempster's Rule of combination can be used to combine information from multiple pieces of evidence (including prior probabilities). Originally, Dempster's Rule of combination is given for BBAs in the following

form:

$$S(\boldsymbol{m}_k) = \frac{1}{1-K} \sum_{\boldsymbol{m}_k = \boldsymbol{m}_i \cap \boldsymbol{m}_j \neq \emptyset} S(\boldsymbol{m}_i) S(\boldsymbol{m}_j) \tag{9.6}$$

$$1 - K = \sum_{\boldsymbol{m}_k = \boldsymbol{m}_i \cap \boldsymbol{m}_j \neq \emptyset} S(\boldsymbol{m}_i) S(\boldsymbol{m}_j) \tag{9.7}$$

where K is a normalization constant, to ensure that the terms $S(\boldsymbol{m}_k)$ sum to unity. In a similar manner, the Generalized Dempster's Rule of Combination is applied to the rows of $\boldsymbol{G}$ that pertain to $\boldsymbol{m}_k$ (or equivalently, $G[\boldsymbol{m}_k, :]$)

$$G_{12}[\boldsymbol{m}_k, :] = \frac{1}{1-K} \sum_{\boldsymbol{m}_k = \boldsymbol{m}_i \cap \boldsymbol{m}_j \neq \emptyset} G_1[\boldsymbol{m}_i, :] \circ G_2[\boldsymbol{m}_j, :] \tag{9.8}$$

$$1 - K = \sum_{\boldsymbol{m}_k = \boldsymbol{m}_i \cap \boldsymbol{m}_j \neq \emptyset} \text{mean}_{x \neq 0}(G[\boldsymbol{m}_i, :] \circ G[\boldsymbol{m}_j, :]) \tag{9.9}$$

where $X \circ Y$ denotes the Hadamard (or element-wise) product between X and Y, while $\text{mean}_{x \neq 0}(X)$ is the mean of the non-zero values of X.

Shortcut combination rule with Bayesian GBBAs

When the Generalized Dempster's rule is applied and at least one of the two GBBAs is Bayesian (having no support to ambiguous modes), the resulting GBBA will also be Bayesian. In such a case, it is easier to apply a shortcut rule that will yield the exact answer with much less computational burden. Here, we can consider the case where $\boldsymbol{G}_1$ is a is a Bayesian prior (expressed as $P_1(M)$), and where $\boldsymbol{G}_2$ is an arbitrary GBBA, the resulting GBBA ($\boldsymbol{G}_{12}$) can be expressed as

$$\boldsymbol{G}_{12}(m_i, m_i) = \frac{1}{1-K} P_1(m_i) In_2(m_i) \tag{9.10}$$

$$\boldsymbol{G}_{12}(m_i, m_{j \neq i}) = 0$$

$$1 - K = \sum_m P_1(M) In_2(M) \tag{9.11}$$

where $In_2(m_i)$ is the inclusive probability of m_i expressed as

$$In_2(m_i) = \sum_{\boldsymbol{m}_k \supseteq m_i} \boldsymbol{G}_2[m_k, m_i] = \boldsymbol{G}_2[:, m_i]^T \boldsymbol{\Theta}^*[:, m_i]$$

The end result is $\boldsymbol{G}_{12}(m_i, m_i)$ being a diagonal matrix, with the main diagonals representing Bayesian posterior probabilities. The shortcut combination rule is particularly useful in a dynamic setting, where successive combinations will yield Bayesian posteriors pretty quickly; it is better to simply use a Bayesian prior in the first step and use the short-cut rule for every successive combination.

9.3 Illustrative Example of Proposed Methodology

9.3.1 Introduction

To demonstrate how to use the Generalized Dempster's rule, we consider the same example as in Chapter 8, the control loop shown in Figure 9.1 where the same modes and evidence are considered

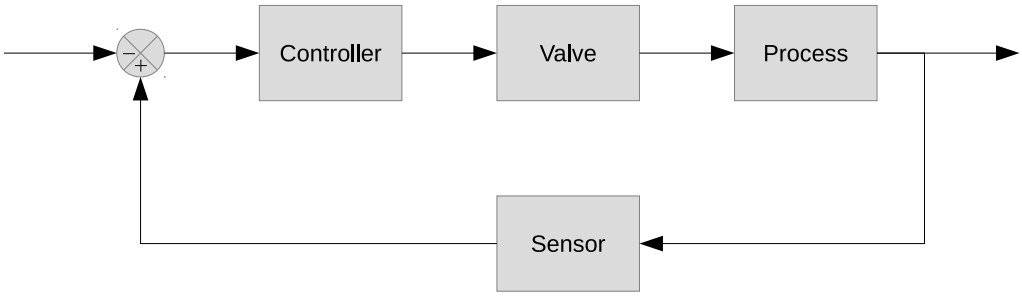


Figure 9.1: Typical control loop

9.3.2 Offline Step 1: Historical data collection

Again, the first step is to go through the historical data and note the instances where each of the four possible modes occurs.

1. m_1 $[0, 0]$ where bias and stiction do not occur
2. m_2 $[0, 1]$ where bias does not occur but stiction does
3. m_3 $[1, 0]$ where bias occurs but stiction does not
4. m_4 $[1, 1]$ where both bias and stiction occur

Data should be collected according to each mode. In certain instances, one of these ambiguous modes may also occur

1. $\{m_1, m_2\}$ $[\times, 0]$ where bias does not occur and stiction is undetermined
2. $\{m_1, m_3\}$ $[\times, 1]$ where bias is undetermined and stiction does not occur
3. $\{m_2, m_4\}$ $[0, \times]$ where bias is undetermined and stiction occurs
4. $\{m_3, m_4\}$ $[1, \times]$ where bias occurs and stiction is undetermined
5. $\{m_1, m_2, m_3, m_4\}$ $[\times, \times]$ where both bias and stiction are undetermined

Such ambiguous cases should be classified under one of these ambiguous modes.

9.3.3 Offline Step 2: Mutual Information Criterion (optional)

As in Chapter 8, this step is optional. One can reduce the dimensionality of the problem by assuming independence where the *mutual information criterion* (MIC) yields a result close to zero (such as less than 0.01).

$$MIC = \begin{bmatrix} 0 & 0 \\ MIC(1,2) & 0 \end{bmatrix}$$

where, for example, $MIC(1,2)$ is calculated as

$$MIC(1,2) = \sum_{E_1} \sum_{E_2} p(E_1, E_2|m) \log \left(\frac{p(E_1, E_2|M)}{p(E_1|M)p(E_2|M)} \right)$$

As in the Second-Order Bayesian method, the MIC matrix must be calculated for every unambiguous mode. For each mode, the two pieces of evidence are either considered independent or dependent. For example, let us say that under Mode (1), the probability is distributed as given in Table 9.1

Table 9.1: Probability of evidence given Mode 1

	$E = [0,0]$	$E = [0,1]$	$E = [1,0]$	$E = [1,1]$
$n(E\|m_1)$	70	10	15	5
$p(E\|m_1)$	0.7	0.1	0.15	0.05

The MIC for Mode 1 is given as

$$\begin{aligned} MIC &= p([0,0]|m_1) \log \left(\frac{p([0,0]|m_1)}{p([0,\times]|m_1)p([\times,0]|m_1)} \right) \\ &\quad + p([0,1]|m_1) \log \left(\frac{p([0,1]|m_1)}{p([0,\times]|m_1)p([\times,1]|m_1)} \right) \\ &\quad + p([1,0]|m_1) \log \left(\frac{p([1,0]|m_1)}{p([1,\times]|m_1)p([\times,0]|m_1)} \right) \\ &\quad + p([1,1]|m_1) \log \left(\frac{p([1,1]|m_1)}{p([1,\times]|m_1)p([\times,1]|m_1)} \right) \\ &= 0.7 \log \left(\frac{0.7}{(0.8)(0.85)} \right) + 0.1 \log \left(\frac{0.1}{(0.8)(0.15)} \right) \\ &\quad + 0.15 \log \left(\frac{0.15}{(0.2)(0.85)} \right) + 0.05 \log \left(\frac{0.05}{(0.2)(0.15)} \right) \\ &= 0.0088259 \end{aligned}$$

Because this number (0.0088259) is so small, the two monitors E_1 and E_2 can be considered independent under this mode.

9.3.4 Offline Step 3: Calculate reference value

Generalized BBAs use inclusive Θ values ($\mathbf{\Theta}^*$) as a reference for Θ in the same manner as the second-order Bayesian method uses the informed values ($\hat{\Theta}$). The boldface notation indicates that $\mathbf{\Theta}^*$ takes the form of a matrix; in this illustration, $\mathbf{\Theta}^*$ takes the following form:

$$\mathbf{\Theta}^* = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \theta^*\{\frac{m_1}{m_1,m_2}\} & \theta^*\{\frac{m_2}{m_1,m_2}\} & 0 & 0 \\ \theta^*\{\frac{m_1}{m_1,m_3}\} & 0 & \theta^*\{\frac{m_3}{m_1,m_3}\} & 0 \\ 0 & \theta^*\{\frac{m_2}{m_2,m_4}\} & 0 & \theta^*\{\frac{m_4}{m_2,m_4}\} \\ 0 & 0 & \theta^*\{\frac{m_3}{m_3,m_4}\} & \theta^*\{\frac{m_4}{m_3,m_4}\} \\ \theta^*\{\frac{m_4}{m_1,m_2,m_3,m_4}\} & \theta^*\{\frac{m_4}{m_1,m_2,m_3,m_4}\} & \theta^*\{\frac{m_4}{m_1,m_2,m_3,m_4}\} & \theta^*\{\frac{m_4}{m_1,m_2,m_3,m_4}\} \end{bmatrix}$$

where each element of $\mathbf{\Theta}^*$ is calculated as

$$\begin{aligned} \mathbf{\Theta}^*[i,j] &= 1 \qquad & m_i \cap \boldsymbol{m}_k \neq \emptyset \\ \mathbf{\Theta}^*[i,j] &= 0 \qquad & m_i \cap \boldsymbol{m}_k = \emptyset \end{aligned}$$

Resulting in

$$\mathbf{\Theta}^* = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

9.3.5 Online Step 1: Calculate support

For this application, we assume that the priors are unambiguous, so that the shortcut method can be applied. When a new piece of evidence $[E_1, E_2]$ becomes available, we calculate the support according to

$$S(E_1, E_2|\boldsymbol{m}_k) = \frac{n(E_1, E_2|\boldsymbol{m}_k)}{n(\boldsymbol{m}_k)}$$

If E_1 and E_2 are considered independent given m_k, the support can be calculated separately

$$S(E_1|\boldsymbol{m}_k) = \frac{n(E_1|\boldsymbol{m}_k)}{n(\boldsymbol{m}_k)} \qquad S(E_2|\boldsymbol{m}_k) = \frac{n(E_2|\boldsymbol{m}_k)}{n(\boldsymbol{m}_k)}$$

The support can be arranged in matrix form as follows

$$\boldsymbol{S} = \begin{bmatrix} S(E|m_1) & 0 & 0 & 0 \\ 0 & S(E|m_2) & 0 & 0 \\ 0 & 0 & S(E|m_3) & 0 \\ 0 & 0 & 0 & S(E|m_4) \\ S(E|m_1,m_2) & S(E|m_1,m_2) & 0 & 0 \\ S(E|m_1,m_3) & 0 & S(E|m_1,m_3) & 0 \\ 0 & S(E|m_2,m_4) & 0 & S(E|m_2,m_4) \\ 0 & 0 & S(E|m_3,m_4) & S(E|m_3,m_4) \\ S(E|m_1,\ldots,m_4) & S(E|m_1,\ldots,m_4) & S(E|m_1,\ldots,m_4) & S(E|m_1,\ldots,m_4) \end{bmatrix}$$

where each column corresponds to an unambiguous mode, and each row corresponds to an ambiguous mode. Note that for some modes (hence columns), evidence independence is assumed, thus some columns have multiple entries to correspond to multiple pieces of evidence.

For example, consider the data collected in Table 9.2 with the support calcualted in Table 9.3.

Table 9.2: Frequency of modes containing m_1

	$E=[0,0]$	$E=[0,1]$	$E=[1,0]$	$E=[1,1]$
$n(E\|m_1)$	70	10	15	5
$n(E\|m_2)$	14	59	6	21
$n(E\|m_3)$	13	7	58	22
$n(E\|m_4)$	12	8	23	57
$n(E\|m_1,m_2)$	25	15	7	3
$n(E\|m_1,m_3)$	20	20	6	4
$n(E\|m_2,m_4)$	19	17	7	7
$n(E\|m_3,m_4)$	19	8	7	16
$n(E\|m_1,m_2,m_3,m_4)$	7	6	6	6

Table 9.3: Support of modes containing m_1

	$E=[0,0]$	$E=[0,1]$	$E=[1,0]$	$E=[1,1]$
$S(E\|m_1)$	0.7	0.1	0.15	0.05
$S(E\|m_2)$	0.14	0.59	0.6	0.21
$S(E\|m_3)$	0.13	0.07	0.58	0.22
$S(E\|m_4)$	0.12	0.08	0.23	0.57
$S(E\|m_1,m_2)$	0.5	0.3	0.14	0.06
$S(E\|m_1,m_3)$	0.4	0.4	0.12	0.08
$n(E\|m_2,m_4)$	0.38	0.34	0.14	0.14
$n(E\|m_3,m_4)$	0.38	0.16	0.14	0.32
$S(E\|m_1,m_2,m_3,m_4)$	0.28	0.24	0.24	0.24

If $E = [0, 1]$ is observed, the support matrix $\boldsymbol{S}$ takes on the following form:

$$\boldsymbol{S} = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.59 & 0 & 0 \\ 0 & 0 & 0.07 & 0 \\ 0 & 0 & 0 & 0.08 \\ 0.3 & 0.3 & 0 & 0 \\ 0.4 & 0 & 0.4 & 0 \\ 0 & 0.34 & 0 & 0.34 \\ 0 & 0 & 0.16 & 0.16 \\ 0.24 & 0.24 & 0.24 & 0.24 \end{bmatrix}$$

9.3.6 Online Step 2: Calculate the GBBA

The GBBA is calculated by taking the derivative of Eqn (9.2), which in vector form is written as

$$p(E|m_i, \boldsymbol{\Theta}) = \frac{[\boldsymbol{S}[:, m_i] \circ \boldsymbol{n}]^T \boldsymbol{\Theta}[:, m_i]}{\boldsymbol{n}^T \boldsymbol{\Theta}[:, m_i]}$$

where $\boldsymbol{n}$ is the vertical vector of mode frequencies.

$$\boldsymbol{n} = [n(m_1), n(m_2), \ldots, n(m_1, m_2, m_3, m_4)]^T$$

In the case of Table 9.2, $\boldsymbol{n}$ is given as

$$\boldsymbol{n} = [100, 100, 100, 100, 50, 50, 50, 50, 25]^T$$

In MATLAB, given the variables `S` (in the form of $\boldsymbol{S}$), `Theta` (in the form of $\boldsymbol{\Theta}$) and `n` (in the form of n), probabilities can be calculated using the following code:

```
n_m = length(S(1,:));
P = zeros(n_m,1);
for m = 1:n_m
    Num = (S(:,m).*n)'*Theta(:,m);
    Den = n'*Theta(:,m);
    P(m) = Num/Den;
end
```

In a similar manner, we can calculate the derivatives as

$$\left.\frac{\partial\ p(E|m_i, \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[k, m_i]}\right|_{\hat{\Theta}} = \frac{\boldsymbol{n}[k]\boldsymbol{S}[k, m_i]}{\boldsymbol{n}^T \boldsymbol{\Theta}^*[:, m_i]} - \frac{\boldsymbol{n}[k]\left(\boldsymbol{n} \circ \boldsymbol{S}[:, m_i]\right)\boldsymbol{\Theta}^*[:, m_i]}{\left(\boldsymbol{n}^T \boldsymbol{\Theta}^*[:, m]\right)^2}$$

which, in MATLAB can be obtained using

```
n = sum(N,2); %Find number of samples for each mode
[n_M,n_m] = size(S);
amb = (n_m+1):n_M; % indices of ambiguous modes
for m = 1:n_m
    Num = (S(:,m).*n)'*Theta(:,m); %Likelihood numerator
```

```
6       Den = n'*Theta(:,m); %Likelihood denominator
7       P(m,1) = Num/Den; %Inclusive Likelihood
8
9       for k = amb % Only consider ambiguous modes for dP
10          if ModeStruct (k,m) == 1
11              %Abridged code which used Num and Den
12              dP(k-n_m,m) = n(k)*S(k,m)/Den - n(k)*Num/(Den^2);
13          end
14      end
15  end
```

After calculating the derivatives, the GBBA will need reference probabilities. The reference probability is calculated as

$$\tilde{p}(E|m_i,\boldsymbol{\Theta}^*) = p(E|m_i,\boldsymbol{\Theta}^*) - \sum_{\boldsymbol{m}_k \supset m_i} \left.\frac{\partial\, p(E|m_i,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[k,m_i]}\right|_{\hat{\Theta}} \boldsymbol{\Theta}^*[k,m_i]$$

As an example, for mode 1, $\tilde{p}(E|m_1,\boldsymbol{\Theta}^*)$ is given as

$$\begin{aligned}\tilde{p}(E|m_1,\boldsymbol{\Theta}^*) = p(E|m_1,\boldsymbol{\Theta}^*) &- \left.\frac{\partial\, p(E|m_1,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[5,1]}\right|_{\hat{\Theta}} \boldsymbol{\Theta}^*[5,1] \\ &- \left.\frac{\partial\, p(E|m_1,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[6,1]}\right|_{\hat{\Theta}} \boldsymbol{\Theta}^*[6,1] - \left.\frac{\partial\, p(E|m_1,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[9,1]}\right|_{\hat{\Theta}} \boldsymbol{\Theta}^*[9,1]\end{aligned}$$

where $\boldsymbol{\Theta}^*$ is the matrix given in Offline Step 3. In MATLAB, the reference probability `Pr` can be obtained using the following code

```
1   for m = 1:n_m
2       Pr(m,1) = P(m,1) - dP(:,m)'*Theta(amb,m);
3   end
```

Using these inputs, the resulting GBBA is given as

$$\boldsymbol{G} = \begin{bmatrix} \tilde{p}(E|m_1,\boldsymbol{\Theta}^*) & 0 & 0 & 0 \\ 0 & \tilde{p}(E|m_2,\boldsymbol{\Theta}^*) & 0 & 0 \\ 0 & 0 & \tilde{p}(E|m_3,\boldsymbol{\Theta}^*) & 0 \\ 0 & 0 & 0 & \tilde{p}(E|m_4,\boldsymbol{\Theta}^*) \\ \left.\frac{\partial\, p(E|m_1,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[5,1]}\right|_{\hat{\Theta}} & \left.\frac{\partial\, p(E|m_2,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[5,2]}\right|_{\hat{\Theta}} & 0 & 0 \\ \left.\frac{\partial\, p(E|m_1,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[6,1]}\right|_{\hat{\Theta}} & 0 & \left.\frac{\partial\, p(E|m_3,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[6,3]}\right|_{\hat{\Theta}} & 0 \\ 0 & \left.\frac{\partial\, p(E|m_2,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[7,2]}\right|_{\hat{\Theta}} & 0 & \left.\frac{\partial\, p(E|m_4,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[7,4]}\right|_{\hat{\Theta}} \\ 0 & 0 & \left.\frac{\partial\, p(E|m_3,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[8,3]}\right|_{\hat{\Theta}} & \left.\frac{\partial\, p(E|m_4,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[8,4]}\right|_{\hat{\Theta}} \\ \left.\frac{\partial\, p(E|m_1,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[9,1]}\right|_{\hat{\Theta}} & \left.\frac{\partial\, p(E|m_2,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[9,2]}\right|_{\hat{\Theta}} & \left.\frac{\partial\, p(E|m_3,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[9,3]}\right|_{\hat{\Theta}} & \left.\frac{\partial\, p(E|m_4,\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}[9,4]}\right|_{\hat{\Theta}} \end{bmatrix}$$

In MATLAB, obtaining the GBBA is a simple matrix concatenation

```
1       GBBA = [diag(Pr);dP]
```

In our example, the GBBA is found to be

$$\boldsymbol{G} = \begin{bmatrix} 0.16 & 0 & 0 & 0 \\ 0 & 0.57 & 0 & 0 \\ 0 & 0 & 0.08 & 0 \\ 0 & 0 & 0 & 0.10 \\ 0.02 & -0.07 & 0 & 0 \\ 0.05 & 0 & 0.10 & 0 \\ 0 & -0.06 & 0 & 0.07 \\ 0 & 0 & 0.02 & 0.01 \\ 0.00 & -0.05 & 0.02 & 0.02 \end{bmatrix}$$

9.3.7 Combine BBAs and diagnose

In this example, the prior probability is unambiguous. Thus the shortcut method can be used for combination. The first step to the shortcut method is to calculate the inclusive probability $P^*(E|M, \boldsymbol{\Theta})$

$$p(E|M, \boldsymbol{\Theta}^*) = \boldsymbol{G}[:, M]^T \boldsymbol{\Theta}^*[:, M]$$

where $\boldsymbol{\Theta}^*$ is the inclusive probability, where all non-zero values are set to 1

$$\boldsymbol{\Theta}^* = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

The posterior probability can be obtained using the following short-cut expression

$$p(M|E) = \frac{1}{K} p(E|M, \boldsymbol{\Theta}^*) p(M) = \frac{1}{K} In(E|M) p(M)$$
$$K = \sum_k p(E|m_k, \boldsymbol{\Theta}^*) p(m_k)$$

After the BBAs are combined, the posterior probability $p(M|E)$ has a single value for each mode (it is not variable with respect to $\boldsymbol{\Theta}$), thus one simply diagnoses the mode based on the highest posterior.

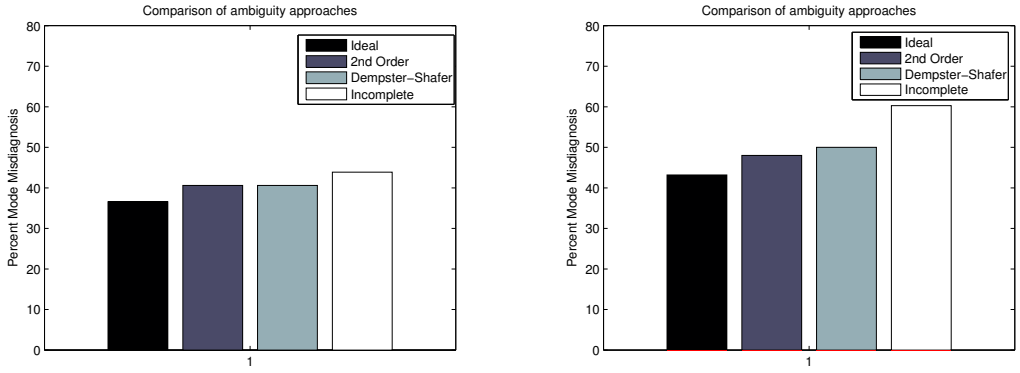
9.4 Simulated Case

The proposed second-order method was tested on the simulated Tenessee Eastman problem. In the same manner as the second-order Bayesian method, data was masked as ambiguous based on its resemblance toward other modes. If 30 % of the data is classified as ambiguous,

then for every mode, it classifies the 70 % of the most likely data for this mode as unambiguous, but each remaining data point will have at least one additional mode associated with it, based on its proximity toward other modes.

Results for the Generalized Dempster-Shafer method are compared to the ideal scenario (where no modes were ambiguous), the incomplete Bayesian method (which ignored ambiguous mode data) and the second-order method. Results in Figure 9.2 compare diagnosis results based on modes, while results in Figure 9.3 compare diagnosis results based on the individual components that make up the modes.

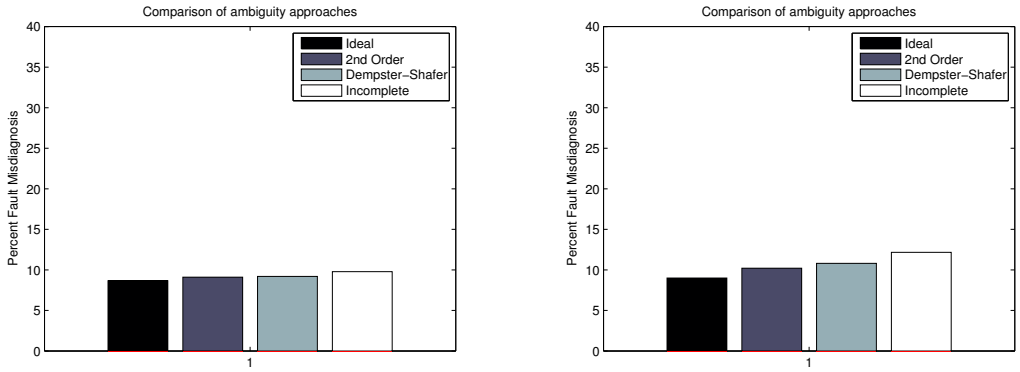
Results for all four scenarios were nearly identical under the case when 30% of the data belonged to ambiguous modes, but more significant improvements were exhibited when 70% of the data came from ambiguous modes. These improvements showed that data was beginning to become more scarce, as performance for the incomplete Bayesian method (where ambiguous mode data was simply ignored) was beginning to decline. Results between the second-order method and the Generalized Dempster-Shafer method were very similar.



(a) 30 % Ambiguous Mode Data

(b) 70 % Ambiguous Mode Data

Figure 9.2: TE mode diagnosis error



(a) 30 % Ambiguous Mode Data

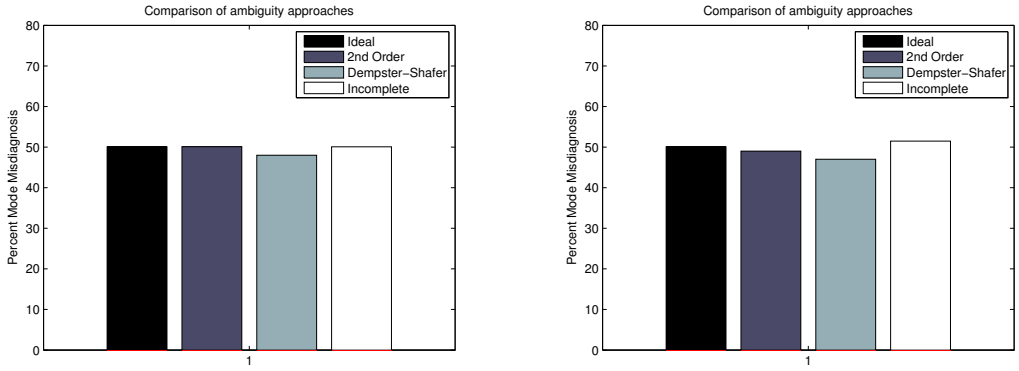
(b) 70 % Ambiguous Mode Data

Figure 9.3: TE component diagnosis error

9.5 Bench Scale Case

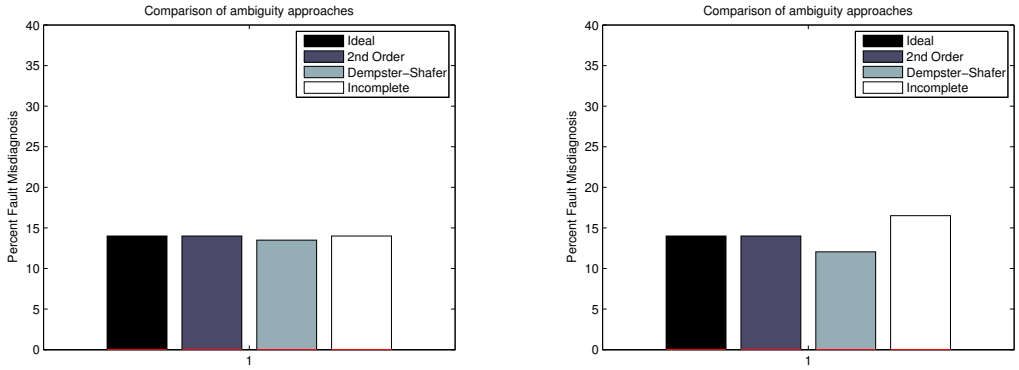
The generalized Dempster-Shafer method was also applied to the lab-scale hybrid tank system. Diagnosis based on modes is shown in Figure 9.4 while diagnosis based on component is shown in Figure 9.5. Performance in this case showed that the generalized Dempster-Shafer method exhibited a slight improvement over the second-order method.

The main difference between the second-order method and the generalized Dempster-Shafer method was that the second-order method distributes ambiguous mode data over the specific modes in accordance to prior probabilities; however, the Generalized Dempster-Shafer method includes all possible ambiguous mode data into each specific mode. If the actual ambiguous mode distribution is significantly different from what one would expect from prior probabilities, the Generalized Dempster-Shafer method will likely yield superior results when compared to the second-order method.



(a) 30 % Ambiguous Mode Data (b) 70 % Ambiguous Mode Data

Figure 9.4: Hybrid tank system mode diagnosis error



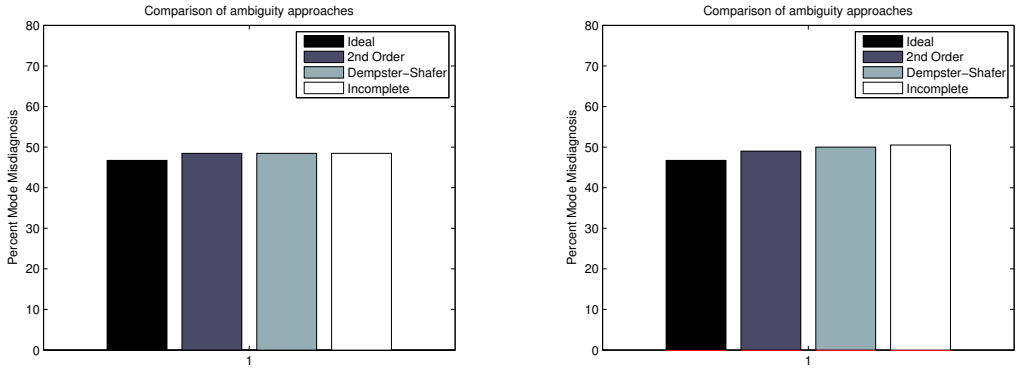
(a) 30 % Ambiguous Mode Data (b) 70 % Ambiguous Mode Data

Figure 9.5: Hybrid tank system component diagnosis error

9.6 Industrial System

Finally, the generalized Dempster-Shafer method was applied to the industrial system alongside the second-order method. In contrast to the experimental system, the second-order method had a slightly superior performance to the generalized Dempster-Shafer method when applied to the industrial system. In this case, the second-order method yields superior results most likely because the real proportions $\mathbf{\Theta}$ are similar to the estimated values $\mathbf{\Theta}^*$ given by the prior probabilities. Results for mode diagnosis are shown in Figure 9.6, while results for component diagnosis are shown in Figure 9.7.

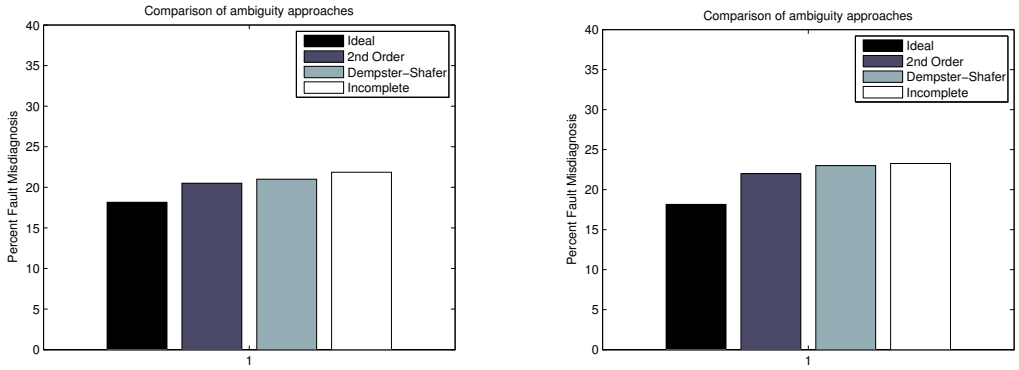
Due to the abundance of data, excluding ambiguous mode data did not have a strong effect when 30% of the data belonged to ambiguous modes, but differences became more pronounced when 70% of the data came from ambiguous modes. There was also a more pronounced change with respect to diagnosing components in contrast to diagnosing modes.



(a) 30 % Ambiguous Mode Data

(b) 70 % Ambiguous Mode Data

Figure 9.6: Industrial system mode diagnosis error



(a) 30 % Ambiguous Mode Data

(b) 70 % Ambiguous Mode Data

Figure 9.7: Industrial system component diagnosis error

Chapter 10

Making use of continuous evidence through kernel density estimation

10.1 Introduction

In previous application chapters, we have considered evidence that takes on a discrete form, such as alarms which were obtained by discretizing a continuous performance metric. By classifying a continuous performance index into low-resolution bins, information is lost. However, using methods that can deal with continuous data preserves this information, and as a result, continuous data methods can significantly improve performance.

The Kernel density estimation method (sometimes called the Parzen-Rosenblatt window method) is a popular method for estimating probability density functions from data. It is a non-parametric method that places a "kernel function" centred around each data point, so that adding the kernel functions results in a smoothed probability density estimate. Kernel density estimation is non-parametric and thus does not assume a predefined shape for the distribution, allowing the distribution estimate to naturally follow the shape of the data distribution, regardless of the shape it takes.

While kernel density estimation does not contain shape parameters, it does contain a crucial smoothing parameter called the bandwidth. A variety of techniques exist for estimating bandwidths, and this chapter will discuss two of the most popular approaches. In addition to the required techniques for kernel densities, there are complementary techniques that can help increase the accuracy of a kernel density estimate. This chapter focuses on how to perform the following techniques:

1. Kernel density estimation
2. Bandwidth selection
3. Dimensionality reduction
4. Handling missing data

10.2 Algorithm

10.2.1 Kernel Density Estimation

The goal of kernel density estimation is to estimate a density function $f(x)$ using kernels $K(x)$ which are centred around each data point in the historical data set D. The kernel $K(x)$ can be any function that integrates to unity; it is preferable for $K(x)$ to be a density function. From a multivariate data set D with n entries, a kernel density estimate from the kernel $K(x)$is obtained using the following sum:

$$f(x) \approx \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|H|^{1/2}} K_H \left(H^{1/2}(x - D_i) \right) \tag{10.1}$$

Due to many desirable mathematical properties, a popular choice of kernel density estimate is the multivariate Gaussian kernel

$$K_H(z) = \frac{1}{\sqrt{(2\pi)^d}} \exp(z^T z)$$

where d is the dimensionality of the data. Using this kernel results in the following kernel density estimate

$$f(x) \approx \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^d |H|}} \exp\left([x - D_i]^T H^{-1} [x - D_i] \right) \tag{10.2}$$

10.2.2 Bandwidth selection

Kernel density estimation is classified as a non-parameteric technique; nevertheless, Kernel density estimates are affected by the parameter H in the same manner that histograms are affected by bin width. When the bin width increases, the histogram becomes smoother, but as the bin size decreases, the histogram takes a rougher shape. Similarly, shrinking the bandwidth H will result in a jagged kernel density estimate, and increasing H will result in a smoother kernel density estimate. Many different methods for selecting bandwidths exist, and is still a somewhat active are of research [78] [79], but the more popular methods are presented in this section.

Optimal bandwidth for normal distributions

Selecting a bandwidth to estimate normal distributions is a mature subject within literature, and the result is well established. Based on the asymptotic mean integrated square error (AMISE) criterion (mentioned in Chapter 6), the optimal bandwidth for estimating a normal distribution with normal kernels is given as

$$H_N = \left(\frac{4}{n(d+2)} \right)^{\frac{2}{d+4}} \Sigma \tag{10.3}$$

where Σ is the covariance of the normal distribution, d is the dimension of the data, and n is the number of sample data points. In practice, the sample covariance S can be used in place of Σ. While this bandwidth is optimal for normal distributions, for many other distributions, this bandwidth can be larger than optimal resulting in an over-smoothed distribution, especially if the distribution has more than one mode. Engineering judgement can be exercised by inspecting the data to see if the shape deviates significantly from normal.

In higher dimensions, the direction of the data can change in different locations. Thus it is often safer to use the main diagonal of the covariance estimate S instead of the full matrix, as S will stretch the covariance matrix in a certain direction. Using the main diagonal will eliminate directional preference.

10.2.3 Adaptive bandwidths

One problem often encountered in kernel density estimation is that when a single bandwidth is used, peaks tend to be over-smoothed, while tails tend to be under-smoothed. Adaptive bandwidth estimation is a common solution for this problem. In the adaptive bandwidth estimation problem, a pilot density function is estimated in order to give a rough probability for all the data points. These probabilities are used to scale the bandwidth. Intuitively, larger probabilities result in narrower bandwidths and smaller probabilities result in wider bandwidths.

In the first step, the pilot density is estimated using the optimal normal bandwidth H_N

$$\hat{f}_H^p(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^d |H_N|}} \exp\left([x - D_i]^T H_N^{-1} [x - D_i]\right)$$

Then, we calculate a geometric mean probability g used as a standardization constant.

$$\log(g) = \frac{1}{n}\sum_{i} \log\left[\hat{f}_H^p(D[i])\right] \tag{10.4}$$

Next, we obtain a bandwidth matrix scalar λ_i for each data point.

$$\lambda_i = \left(\frac{\hat{f}_H^p(D[i])}{g}\right)^{\alpha} \tag{10.5}$$

where the parameter α is a user-defined parameter. For practical purposes, it is most often set to $\alpha = 0.5$ for moderate sample sizes (such as 100 sample points in the univariate case), but should be reduced for larger sample sizes. The parameter λ_i is used to scale the bandwidth as follows:

$$H_i = \lambda_i^{-2/d} H_N \tag{10.6}$$

A possible adaptation to this step is to use a local covariance estimate S_i to account for the local direction of the data. In this adaptation, we obtain a sample covariance estimate S_i

using the data closest to $D[i]$ (for example, the closest 25% of the data, or use a weighting function for the covariacne estimator) and scale it according to the following expression:

$$H_i = \lambda_i^{-2/d} \frac{|H_N|^{1/d}}{|S_i|^{1/d}} S_i \tag{10.7}$$

Finally, the kernel density estimate is given as

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^d |H_i|}} \exp\left([x - D_i]^T H_i^{-1} [x - D_i]\right) \tag{10.8}$$

10.2.4 Optional Step: Dimension reduction by multiplying independent likelihoods

Kernel density methods, like discrete methods, tend to suffer performance degradation when dimensionality is increased. For the discrete method, estimation difficulty increases exponentially with respect to the number of monitoring inputs. For example, if 2 monitors were used, each having 3 different states, there would be $2^3 = 9$ bins required for estimation. Now if 15 monitors were used, the number of bins for estimation would balloon to over 14 million. Kernel density methods are more efficient at approximating densities, and thus difficulty may not grow as fast as discrete/histogram methods, but dimensionality can still be a problem.

If monitors are highly correlated, data tends to exhibit lower-dimensionality behaviour, but independent monitors are more problematic. However, independent monitors have a convenient solution. If certain monitors (or groups of monitors) can be considered independent, $E_1 \perp E_2$, then their higher-dimensional joint likelihood $p(E_1, E_2|M)$ can be calculated by multiplying the lower-dimensional individual likelihoods $p(E_1|M)$, $p(E_2|M)$ together.

$$p(E_1, E_2|M) = p(E_1|M)p(E_2|M)$$

Assuming independence and verifying these assumptions through the MIC was previously discussed in Chapter 8. The procedure largely remains the same in this chapter, except that now probability functions $p(\pi_1), p(\pi_2), p(\pi_1, \pi_2)$ can be expressed as kernel density estimates

$$I(x_1; x_2) = \int_{x_1} \int_{x_2} p(x_1, x_2) \log \left(\frac{p(x_1, x_2)}{p(x_1)p(x_2)} \right) \, dx_1 \, dx_2 \tag{10.9}$$

Because the integration is in only one or two dimensions, the integration can be performed using quadrature (for example, via the MATLAB command "quad2d" performs two-dimensional quadrature integration). When the MIC is applied to kernel density estimates, values less than 0.05 are considered negligible, while values greater than 0.2 are considered significant.

10.2.5 Optional Step: Creating independence via Independent Component Analysis

Independent component analysis (ICA) assumes that data was generated by a linear combination of independent variables; it is similar to principal component analysis (PCA) except that the latent variable t is not restricted to be Gaussian. Thus, ICA can be effectively performed on data that is not multivariate normal. Nevertheless, there still is a restriction in that observations y are assumed to be linear combinations of latent variables f

$$y - \mu = At$$

The goal of ICA is to obtain the transformation matrix A, or more importantly, its inverse $W = A^{-1}$. While many algorithms for ICA exist (some are even based on the MIC), the fixed-point algorithm has shown to exhibit both accuracy and speed. Software for this algorithm has been previously developed under the MATLAB and Octave platforms and is available at the following website: http://research.ics.aalto.fi/ica/fastica/.

Independent component analysis has been shown to be effective when observation inputs consist of raw or relatively unprocessed data. However, if the data consists of monitors that were tuned to be sensitive to underlying problem sources, ICA can reduce this sensitivity and possibly result in poorer diagnostic performance.

10.2.6 Optional Step: Replacing missing values

When values are missing for a kernel density estimation, one cannot use the marginalization methods that are used for discrete monitors. However, being able to assume independence among smaller groups of observations makes missing values less of an issue. For example, consider a system with nine observations $y_1, y_2, \ldots, y_9$. If it was possible to break them down into three independent groups $[y_1, y_2, y_3] \perp [y_4, y_5, y_6] \perp [y_7, y_8, y_9]$, then if y_7 is missing, we would only have to discard y_8 and y_9; if independence was not assumed, then observations $y_1, \ldots, y_6$ would also have to be discarded along with y_8 and y_9.

If discarding missing data significantly reduces data reliability, then missing data entries have to be estimated. This can be done by applying the following steps:

1. Estimate the kernel density function using only complete data entries
2. Estimate the expected values of the incomplete data entries
3. Using the estimated data entries, estimate the kernel density function which includes the new data points
4. Repeat the log likelihood maximization based on the new kernel density function
5. Repeat steps (3) and (4) until the likelihood converges

This approach resembles the EM algorithm, except that the missing values must be explicitly calculated instead of its statistics. Such an approach is required because the kernel density function is non-parametric.

After obtaining the kernel density estimate, one can use *kernel density regression* to replace the missing values. There are two main techniques for kernel density regression: the *zeroth-order (Nadaraya-Watson) method*, and the *first-order method*.

1. **Zeroth-order (Nadaraya-Watson) method:** For a point with known elements x and missing elements y, we look at the historical record of data having both known X and Y values. The value of y is calculated by a weighted average of the historical Y values, weighted based on the proximity of their X components to X.

$$E(y) = \frac{\sum_{i=1}^{n} K\left(H_x^{-1/2}[X_i - x]\right) Y_i}{\sum_{i=1}^{n} K\left(H_x^{-1/2}[X_i - x]\right)} \tag{10.10}$$

2. **First-order method:** The zeroth-order method tends to have a flat bias of the function, particularly around peaks in y and around the edges of the data. In order to correct this bias, the first-order method is proposed. Instead of taking a weighted average of Y, weighted linear regression on Y is performed in stead. Given the point x, y with known elements x and unknown elements y, we first obtain the regressor variable Z as

$$Z_i = \begin{bmatrix} 1 \\ X_i - x \end{bmatrix}$$

The weighted linear regression is then performed as

$$\begin{bmatrix} E(y) \\ \hat{\beta}(x) \end{bmatrix} = \left[\sum_{i=1}^{n} K\left(H_x^{-1/2}[X_i - x]\right) Z_i Z_i^T\right]^{-1} \left[\sum_{i=1}^{n} K\left(H_x^{-1/2}[X_i - x]\right) Z_i Y_i\right] \tag{10.11}$$

The result of the regression is a vector, with $E(y)$ as the first element (corresponding to $Z_i = 1$), with the remaining elements $\hat{\beta}(x)$ being ignored.

After finding $E(y)$ for all missing data points, the completed data points can be added to the data. For a second iteration, because estimates are obtained for missing values, one can use the kernel density estimate from the completed data history to estimate $E(y)$ again. The steps of updating the kernel density estimate and filling in missing values can be repeated until the likelihood converges.

10.3 Illustrative Example of Proposed Methodology

Again, we consider the example presented in Chapter 8, the control loop shown in Figure 10.1 where the same modes and evidence are considered The methodology of kernel density estimation is data-intensive and cannot be easily summarized; thus the example will be illustrated through MATLAB/Octave code.

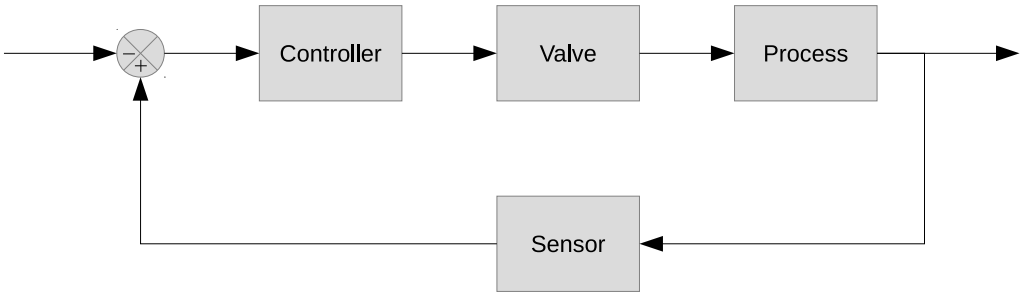


Figure 10.1: Typical control loop

Kernel Density Estimation functions

The code below is an optimized MATLAB/Octave implementation of kernel density estimation which works in arbitrary dimensions. The main feature is that Cholesky decomposition is used for the bandwidth so that it only needs to be performed once offline.

```
function DensityInfo = fKernelEstimateNorm(X)
[n,d] = size(X);
%A diagonal covariance matrix tends yield more stable results
S = diag(var(X));
%Optimal bandwidth for Gaussian estimation
H = (4/(n*(d+2)))^(2/(d+4))*S;

%Inversion from cholesky decomposition (for speed and forced symmetry)
HRi = eye(d)/chol(H);

%we use the log of normalization constant for the kernel density to avoid NaNs
BWMd = det(H);
k = -0.5*log( (2*pi)^d*BWMd ) - log(n);

DensityInfo.bwm = H;        %bandwidth matrix
DensityInfo.bwmZt = HRi';    %transformation to Z
DensityInfo.logk = k;    %log of KDE normalizing constant
DensityInfo.data = X;       %Data
```

The `fKernelEstimateNorm` command is used to set up the bandwidth parameters in a way to allow efficient kernel density estimation from the `fKernelDensity` given as

```
function fx = fKernelDensity(x,DensityInfo)
%x is multivariate random variable realization (row vector)
%multiple rows in x will result in multiple probabilities

Data = DensityInfo.data;        %KDE data
k(1,:) = DensityInfo.logk;    %Log of KDE normalization
Zt = DensityInfo.bwmZt;     %Transformation to Mahalanobis distance


%This function estimates multiple probabilities from multiple rows in x
%Data has nD rows, each are horizonal entries of x
[nX,p] = size(x);
```

```
13  [nD,¬] = size(Data);
14
15  %Calculate kernel density (can take both static and variable kernels)
16  fx = zeros(nX,1);
17  for n = 1:nX
18      Xd = ( Data - ones(nD,1)*x(n,:) )'; %Raw distances
19      DM = sum( (Zt*Xd).^2,1);    %Mahalanobis distances
20      Pe = k - 0.5*DM;    %Exponent of probability
21      fx(n,1) = sum(exp(Pe));    %Probability density
22  end
```

This function treats `x` as a set of row vectors, so that multiple rows will yield multiple likelihoods. Note that the log normalization constant is added in the exponent; this is less likely to result in numerical errors if `k` and `DM` are large. For the sake of speed, it is key to ensure that the exponent `exp(Pe)` is evaluated for the entire vector `Pe` as this is much faster than evaluating elements of `Pe` one at a time (due to parallel computation methods invoked by MATLAB and Octave for vectors).

10.3.1 Offline Step 1: Historical data collection

As in previous cases, the first step is to go through the historical data and note the instances where each of the four possible modes occurs.

1. m_1 $[0, 0]$ where bias and stiction do not occur
2. m_2 $[0, 1]$ where bias does not occur but stiction is does
3. m_3 $[1, 0]$ where bias occurs but stiction does not
4. m_4 $[1, 1]$ where both bias and stiction occur

In this chapter, we assume that no ambiguous modes exist, thus Bayesian methods are used. Nevertheless, because the only aspect that changes is how the likelihood function $p(E|M)$ is calculated (as kernel density estimates are used) kernel density estimates can be combined with the second-order Bayesian method or the Dempster-Shafer method in order to handle ambiguity.

If we consider the MATLAB cell variable `Data` which contains data for each mode, we can set up the kernel density estimates with the following code:

```
1      for m = 1:length(Data)
2          KDE(m) = fKernelEstimateNorm(Data{m})
3      end
```

Offline Step 2: Replacing missing values (optional)

If elements of E are missing, they cannot be used for learning unless the missing values are replaced. The data is separated into two sections $\mathcal{D}_m = [\mathcal{D}_c, \mathcal{D}_{ic}]$. In MATLAB, it is

assumed that missing elements are represented by the value `NaN`; for example

$$\mathcal{D}_m = \begin{bmatrix} 4.5 & 2.8 & 0.3 & 1.2 \\ 2.7 & \texttt{NaN} & 1.0 & 0.9 \\ 5.6 & 3.0 & 0.6 & 0.11 \\ 4.5 & 2.7 & 0.4 & 1.3 \\ 3.6 & 2.8 & 0.5 & \texttt{NaN} \end{bmatrix}$$

Each row represents a historical piece of evidence; from this example, rows 2 and 5 would be moved to the set D_{ic} while rows 1, 3, and 4 would be moved to the set D_c. This sorting can easily be done in MATLAB; consider the data set for mode M denoted by `Data{m}` that has some `NaN` entries in various rows. Sorting is done using the following code:

```
vNaN = sum(Data{m},2); %rows with NaN will sum to NaN
indIC = find(isnan(vNaN)); %identify rows with NaN
indC = find(¬isnan(vNaN)); %identify rows without NaN

%Place incomplete and complete data into appropriate data sets
Dc = Data{m}(indC,:);
Dic = Data{m}(indIC,:);
DcF = Dc;
DicF = Dic;
```

where "¬" is the "not" operator, written as "∼" in MATLAB. Note `DcF` represents the data with missing sets filled in, but on the first iteration, the missing data are not filled in. For each incomplete data entry, obtain the expected value of the missing entries using Kernel density regression.

```
for i = 1:size(Dic,1)
    dic = Dic(i,:);
    Xind = find(¬isnan(dic));
    Yind = find(isnan(dic));
    KDE = fKernelEstimateNorm(DcF(:,Xind));

    dic(i,Yind) = fKernelRegFirst(dic(Xind),DcF(:,Yind),KDE)
    DicF(i,:) = dic;
end
```

The function `fKernelRegFirst` returns the expected value of the missing elements `Yind` using the historical values of the missing component `DcF`(:, `Yind`) and the kernel density estimate `KDE` (obtained using historical values of the available component `DcF`(:, `Xind`)). While the basics of the function `fKernelRegFirst` have been already described, a number of safeguards and efficiency-increasing steps are included making the code quite lengthy, thus details `fKernelRegFirst` are placed in Appendix A. Once estimation has been performed over all elements in `Dic`, one can add completed elements to the data.

```
DcF = [Dc,DicF]
```

Then, one can estimate missing elements `Dic` again with the new completed `Dc` dataset.

10.3.2 Offline Step 3: Mutual Information Criterion (optional)

As in Chapters 8 and 9, this step is optional. The kernel density estimate is used to obtain the probability terms $p(E_1|M)p(E_2|M)$ and $p(E_1, E_2|M)$; if MIC(1,2) is greater than some threshold (0.01), it would be better to assume that E_1 and E_2 were dependent.

$$MIC = \begin{bmatrix} 0 & 0 \\ MIC(1,2) & 0 \end{bmatrix}$$

where $MIC(1,2)$ is calculated as

$$MIC(1,2) = \int_{E_1} \int_{E_2} p(E_1, E_2|m) \log \left(\frac{p(E_1, E_2|M)}{p(E_1|M)p(E_2|M)} \right) dE_1 \; dE_2$$

In MATLAB, the MIC can be calculated using numerical integration (the authors used the two-dimensional quadrature command `quad2d` in MATLAB. Consider the example above, where D is the historical record of the bivariate random variable E. In MATLAB, the mutual information term can be expressed as

```
function MI = fMI(X,Y,KDE,KDEx,KDEy)
   %X and Y are usually scalars, but quad2d prefers the ability to have matrix inputs/outputs
    [ni,nj] = size(X);
    MI = zeros(ni,nj);
    for i = 1:ni
        for j = 1:nj
            p12 = fKernelDensity([X(i,j),Y(i,j)],KDE);
            p1 = fKernelDensity(X(i,j),KDEx);
            p2 = fKernelDensity(Y(i,j),KDEy);
            MI(i,j) = p12*log( p12/(p1*p2) );
        end
    end
end
```

Note, if there are regions where `p12`, `p1`, `p2` are very small (and approach zero), there runs a risk that `MI` will be undefined. It is useful to note that, due to the logarithmic term, if `p12` approaches zero, then `MI` also approaches zero; thus, for reliability, statements made in the code should be inserted to reflect this fact.

The mutual information criterion is obtained by integrating over the aforementioned mutual information term

```
function MIC = fMIC(D)
    x = D(:,1);
    y = D(:,2);

    %Kernel density for univariate and bivariate data sets
    [KDE] =  fKernelEstimateNorm(D);
    [KDEx] = fKernelEstimateNorm(x);
    [KDEy] = fKernelEstimateNorm(y);

    %Set integration boundaries
    minx = min(x) - 0.1*std(x);
    maxx = max(x) + 0.1*std(x);
```

```
13      miny = min(y) - 0.1*std(y);
14      maxy = max(y) + 0.1*std(y);
15
16      %hMI is a function to be integrated
17      hMI = @(X,Y) fMI(X,Y,KDE,KDEx,KDEy);
18
19      %Integrate to obtain MIC
20      MIC = quad2d(hMI,minx,maxx,miny,maxy)
21
22  end
```

The mutual information criterion can be arranged in a lower triangular matrix for grouping

```
1       MICmatrix = zeros(length{Data{m});
2       ne = length(Data{m}(1,:)); %number of evidence sources
3       for j = 1:ne
4           for i = (j+1):ne
5               MICmatrix(i,j) = fMIC(Data{m}(:,i,j))
6           end
7       end
```

When grouping, one prioritizes the elements in the MIC matrix which have large values. If the value in the MIC matrix less than 0.05, the corresponding two pieces of evidence (by row and column) can be considered independent.

10.3.3 Offline Step 4: Independent Component Analysis (optional)

If process data were used directly, applying ICA to transform the data may be advantageous. The fixed-point algorithm [76] can be applied in order to solve for A in the expression

$$y = At + \varepsilon$$

so that the new monitor inputs π^* are calculated as

$$D_m^* = A^{-1} D_m = W D_m$$

The transformed data D_m^* can be used as input instead of the original data D_m.

Using the fast ICA package [76], the MATLAB code is used to produce the information necessary for ICA, that is, the mean and the transformation matrix W.

```
1   Transform.mean = mean(data);
2   [¬,¬,W] = fastica(data','stabilization','on');
3   Transform.W = W;
```

The arguments $'$`stabilization`$'$,$'$`on`$'$ help the algorithm achieve more reliable results.

10.3.4 Offline Step 5: Obtain bandwidths

There are two main options for bandwidth matrices; the optimal bandwidth for Gaussian distributions and the adaptive optimal bandwidth. The first option is applied to all data

points, and for each mode, the bandwidth matrix is given a value of

$$H_m = \left(\frac{4}{n(d+2)}\right)^{\frac{2}{d+4}} \Sigma_m \tag{10.12}$$

so that the kernel density estimate is given as

$$\hat{f}^p_{H_m}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^d |H_m|}} \exp\left([x - D_i]^T H_m^{-1} [x - D_i]\right)$$

where n is the number of data points in $\mathcal{D}_m$, D_i is the i^{th} data point in $\mathcal{D}_m$, d is the dimension of the data in $\mathcal{D}_m$ and Σ_m is the covariance matrix of $\mathcal{D}_m$. The equivalent bandwidth selection step in MATLAB was already given as

```
1      for m = 1:length(Data)
2          KDE(m) = fKernelEstimateNorm(Data{m});
3      end
```

so that likelihood probabilities given x are given as

```
1  for m = 1:length(DATA)
2     L(m) = fKernelDensity(x,KDE(m));
3  end
```

As an alternative to the optimal normal bandwidth, one can use the adaptive bandwidths H_m^k which are different for every k^{th} element.

$$\log(g) = \frac{1}{n}\sum_i \log\left[\hat{f}^p_H(D_i)\right]$$

$$\lambda_i = \frac{\hat{f}^p_H(D_i)}{g}$$

$$H_m[i] = \lambda_i^{-1} H_m$$

so that the kernel density estimate is given as

$$\hat{f}^p_{H_m}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^d |H_m[i]|}} \exp\left([x - D_i]^T H_m[i]^{-1} [x - D_i]\right)$$

When implementing on MATLAB, new commands have to be created:

- `fKernelEstimateNorm`
- `fKernelDensityAdaptive`

The adaptive bandwidth selector is given as

```
1  function [DensityInfo] = fKernelEstimateAdaptive(X)
2
```

```
%Generate initial bandwidth matrix using optimal normal method
[n,p] = size(X);
KDE = fKernelEstimateNorm(X);
H0 = KDE.bwm;

%obtain scaling parameters Lam
fx = fKernelDensity(X,KDE);
g = exp( mean( log(fx)) );
Lam = fx/g;

%Scale bandwidth for individual points
for i = 1:n
    BWMi = H0/Lam(i);
    BWM(:,:,i) = BWMi;
    Zt(:,:,i)  = chol(BWMi,'lower')\eye(p);
    dBWM(i,:)  = det(BWMi);
end
%log of normalizing constant
logk = -0.5*log((2*pi)^p*dBWM) - log(n);

%Save results as a structure
DensityInfo.bwm = BWM;
DensityInfo.bwmZt = Zt;
DensityInfo.logk = logk;
DensityInfo.data = X;

end
```

while the adaptive kernel density estimator is given as

```
function fx = fKernelDensityAdaptive(x,DensityInfo)
%Unload key parameters
Data      = DensityInfo.data;
Zt        = DensityInfo.bwmZt;
k(1,:)    = DensityInfo.logk;

%Data has nD rows, each are horizonal entries of x
[nX,p] = size(x);
[nD,~] = size(Data);

%Calculate kernel density (can take both static and variable kernels)
fx = zeros(nX,1);
kz(1,1,:) = k;
for n = 1:nX
    %find distances (distances are nD vertical vectors)
    Xd = (Data - ones(nD,1)*x(n,:))';
    %Multiply each transformation by the appropriate distance
    for zi = 1:nD
            Z(:,:,zi) = Zt(:,:,zi)*Xd(:,zi);
    end
    Pe(:,1) = kz - 0.5*sum(Z.^2,1); %exponent of probability
    fx(n,1) = sum(exp(Pe));
end
```

If likelihoods contain multiple independent groups, one has to construct the groups first, for example, let us say that under mode 1, the independent groups are $[1, 3, 4]$ and $[2, 5]$ but for mode 2, all evidence is dependent.

```
groups1{1} = [1,3,4];
```

```
groups1{2} = [2,5];
Mgroups{1} = groups1;

groups2{1} = [1,2,3,4,5];
Mgroups{2} = groups2;
```

The optimal normal bandwidths are then obtained as

```
cKDE = cell(length(Mgroups),1);

for m = 1:length(Mgroups)
    groups = Mgroups{m};
    for c = 1:length(groups)
        %use only data from relevent groups for each KDE
        cKDE{m}(c) = fKernelEstimateNorm( Data{m}(:,groups{c}) );
    end
end
```

Note, that one could also use adaptive bandwidths here.

10.3.5 Online Step 1: Calculate likelihood of new data

When this method is performed online, new evidence from monitors E is used as input to calculate the probability. When kernel density estimation is applied

$$p(E|M) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^d |H_m|}} \exp\left([e - D_i]]^T H_m^{-1}[e - D_i]\right)$$

where $H_m[i]$ is used instead of H_m if the adaptive kernel density estimation method is used. Again, probabilities can be calculated using the fKernelDensity command

```
for m = 1:length(KDE)
   L(m) = fKernelDensity(x,KDE(m));
end
```

Note that if the evidence is separated into independent components, there is a likelihood $p(E_c|M)$ for every component c. These component likelihoods have to be multiplied together in order to obtain the net likelihood.

```
for m = 1:length(Data)
    groups = Mgroups{m};
    L(m) = 1;
        for c = 1:length(groups)
            L(m) = L(m)* fKernelDensity( x(groups{c}),KDE{m}(c) );
        end
    end
end
```

10.3.6 Online Step 2: Calculate posterior probability

The posterior probability is obtained by combining the likelihood with a prior according to Bayes' Rule

$$p(M|E) = \frac{p(E|M)p(M)}{\sum_k p(E|M_k)p(m_k)}$$

Note that if there is more than one component,

$$p(E|M) = p(E_1|M) \times p(E_2|M) \times \ldots \times p(E_{n_c}|M)$$

In MATLAB, it is convenient to set the prior $p(\boldsymbol{m}) = \mathrm{P}$ as a vertical vector along with the likelihood $p(E|\boldsymbol{m}) = \mathrm{L}$. In this way, the vector of posteriors can be calculated as

```
1       P = (L.*P)/(L'*P);
```

10.3.7 Online Step 3: Make a diagnosis

The diagnosis is made by selecting the mode with the highest posterior probability.

10.4 Simulated Case

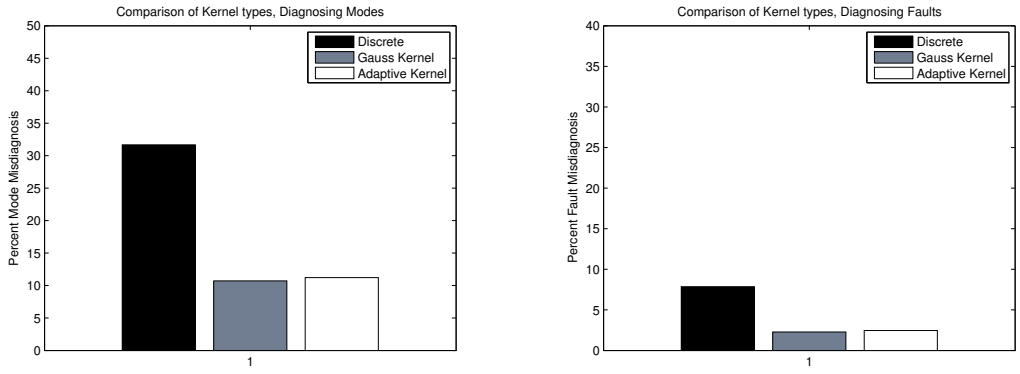
The Kernel density estimation method is used on the Tennessee Eastman problem, a popular benchmark simulation system. As done in previous applications, each fault (or problem source) is simulated, with some of the data being used for learning, while other parts of the data are used for validation. The first set of results (in Figure 10.2) compare diagnosis results from the discrete method against the optimal normal and adaptive kernel density estimation methods. From this figure, kernel density estimation performs significantly better than the discrete methods both for diagnosing modes and problem sources. However, adaptive and optimal Gaussian kernel density estimation methods have similar performance to each other.

In addition, we consider how different grouping methods affect diagnosis rates. Four different grouping approaches are considered.

1. The ***lumped*** approach does not assume independence, and considers all evidence as a single multivariate variable.

2. The ***indepdnent*** approach is the complete opposite; it assumes that all pieces of evidence are independent so that likelihoods are calculated for each piece of evidence and the joint likelihood is combined by multiplying results together.

3. The ***grouped*** approach is a compromise between the lumped and independent approaches. It uses the MIC as an indicator to determine which variables are dependent and which ones are not. The grouping algorithms mentioned in this chapter (as well as



(a) Mode Misdiagnosis Rate

(b) Source Misdiagnosis Rate

Figure 10.2: Tennessee Eastman, discrete vs. KDE

Chapters 8 and 9) are used to separate evidence into independent groups. Likelihoods are calculated for each group and are multiplied together in order to obtain a joint likelihood.

4. Finally, the ICA ***transformed*** method is used to transform the evidence into independent components, so that likelihoods are obtained for each component; the joint likelihood is again obtained through multiplying individual likelihoods together.

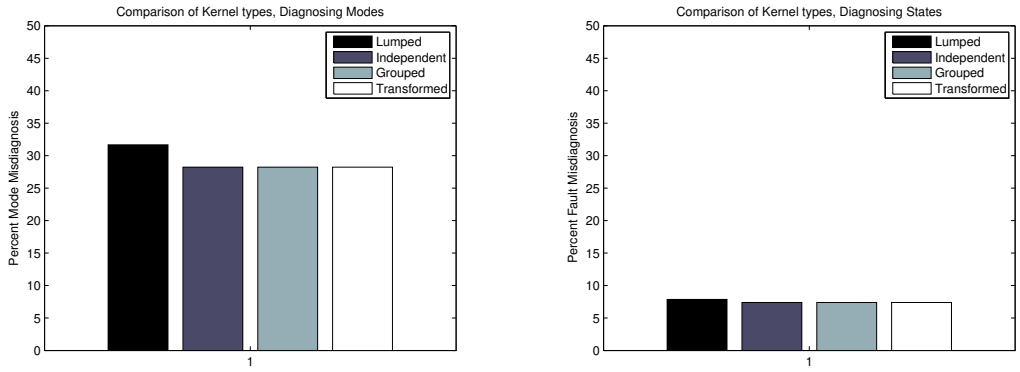
Chapters 8 and 9 have not compared grouping approaches for the discrete method, thus we compare grouping approaches for the discrete method first. The grouping comparison for the discrete method is shown in Figure 10.3. The grouping approaches are also compared for the kernel density estimation method; the comparison for kernel density estimation is shown in Figure 10.4. Because performance of optimal Gaussian and adaptive kernel density estimation is similar, the optimal Gaussian method is used here as it is the simpler of the two.

From Figures 10.3 and 10.4 one can see that different grouping techniques have a negligible effect on diagnosis performance regardless as to whether discrete or kernel density estimation is used.

10.5 Bench Scale Case

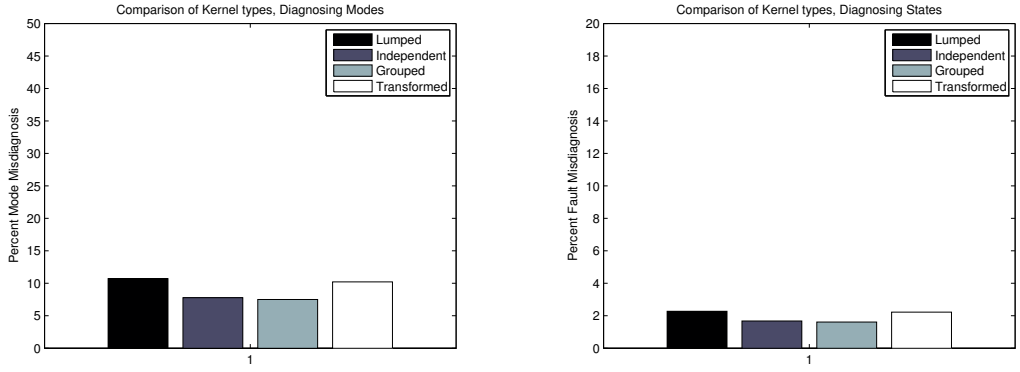
The kernel density estimation method was also used on the hybrid tank system and compared with discrete results. Comparison between discrete and kernel density estimation methods is shown in Figure 10.5. Again, for both mode and component diagnosis, kernel density methods exhibit a very significant improvement over the discrete method. In addition, the adaptive and optimal Gaussian kernel density estimation methods perform similarly yet again.

Grouping methods for the lab scale system are also compared. As seen in Figure 10.3,



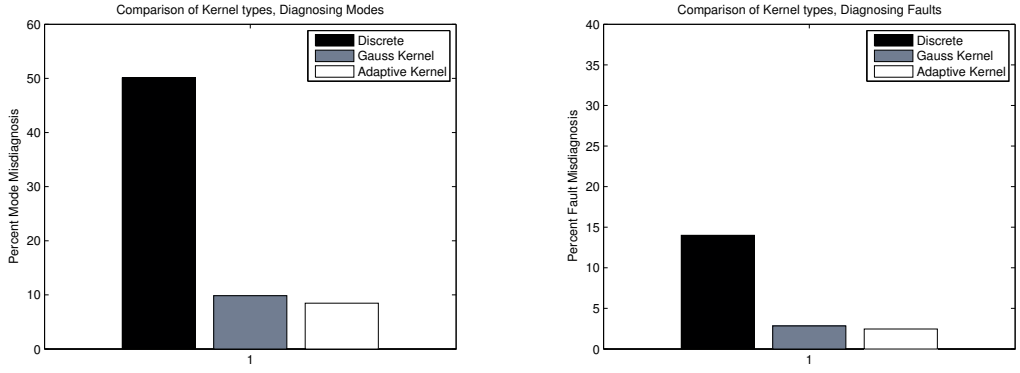
(a) Mode Misdiagnosis Rate (b) Source Misdiagnosis Rate

Figure 10.3: Grouping approaches for discrete method



(a) Mode Misdiagnosis Rate (b) Source Misdiagnosis Rate

Figure 10.4: Grouping approaches for KDE method

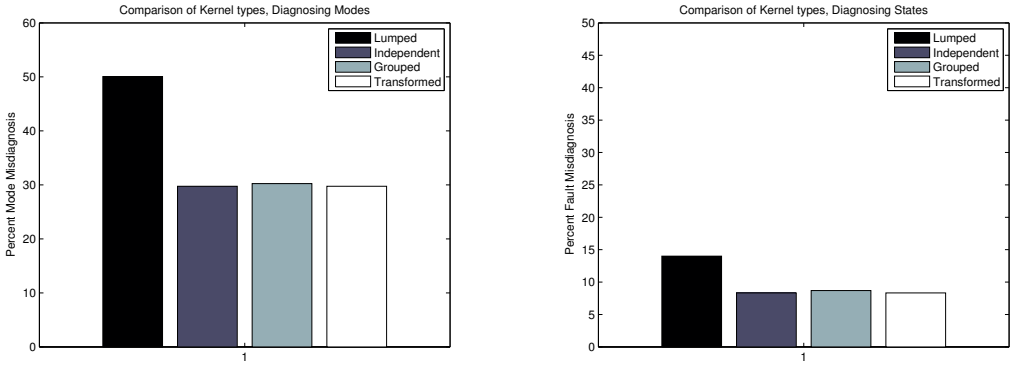


(a) Mode Misdiagnosis Rate (b) Source Misdiagnosis Rate

Figure 10.5: Hybrid tank, discrete vs. KDE

for the discrete method, the independent, grouped and transformed approaches all exhibit a significant improvement over the lumped method. In contrast however, for the kernel

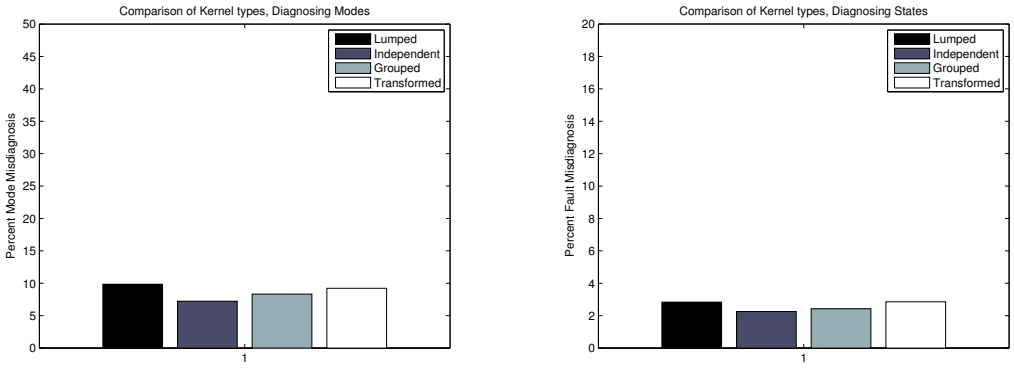
density estimation method, all grouping approaches perform similarly (as seen in Figure 10.4).



(a) Mode Misdiagnosis Rate

(b) Source Misdiagnosis Rate

Figure 10.6: Grouping approaches for discrete method



(a) Mode Misdiagnosis Rate

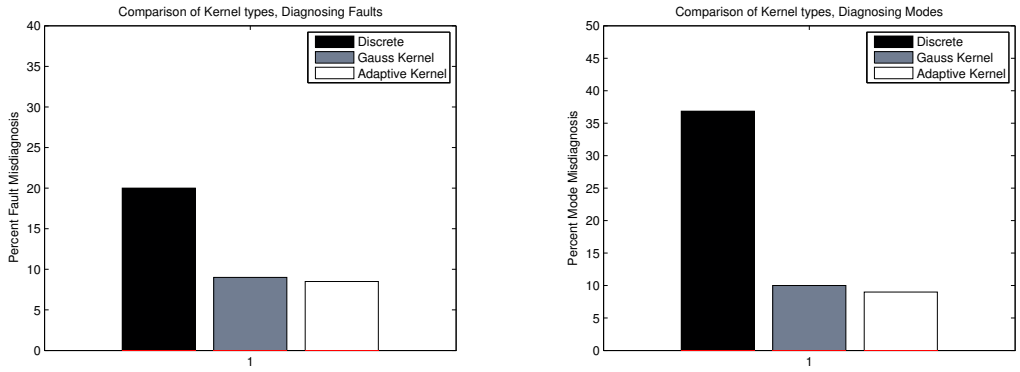
(b) Source Misdiagnosis Rate

Figure 10.7: Grouping approaches for KDE method

10.6 Industrial Scale Case

Finally, the kernel density methods were compared to the discrete methods for the industrial system. In Figure 10.8, yet again we see a significant improvement when kernel density estimation methods are used instead of discrete methods.

Different grouping approaches were tried on the discrete method when applied to the industrial system and results are shown in Figure 10.9. Here, the independent, grouped and transformed approaches outperformed the lumped approach; because process measurements were directly used and a large number of instruments were applied, the transformed approach performed slightly better than the other methods. However, it should be noted that the transformed method is application specific and can result in worse performance

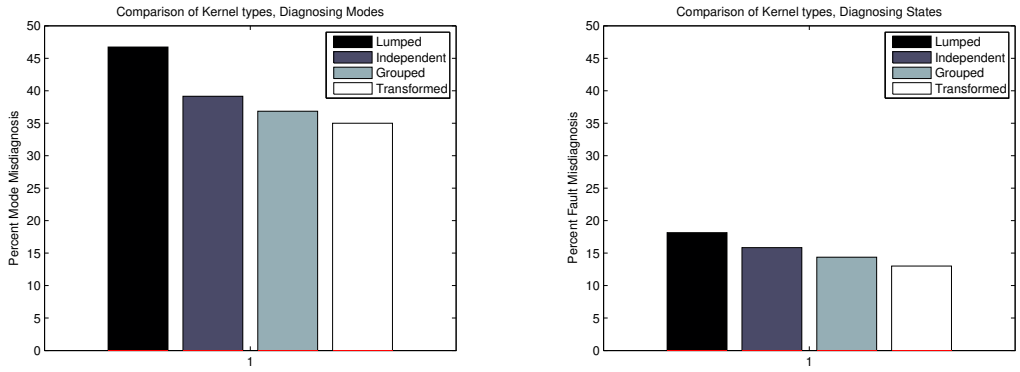


(a) Mode Misdiagnosis Rate

(b) Source Misdiagnosis Rate

Figure 10.8: Solids handling, discrete vs. KDE

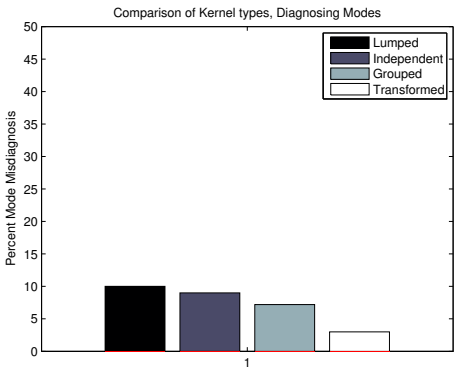
for applications that it is not well suited for. When the different grouping methods were applied to the optimal Gaussian kernel density estimation method, the other approaches still exhibit superior performance over the lumped method.



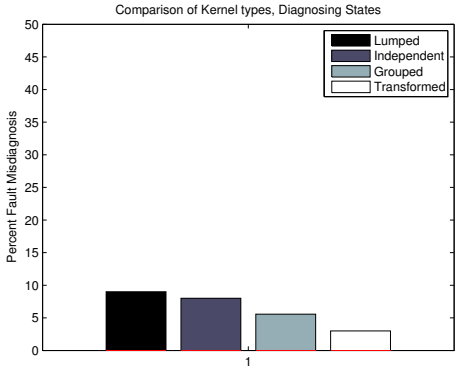
(a) Mode Misdiagnosis Rate

(b) Source Misdiagnosis Rate

Figure 10.9: Grouping approaches for discrete method



(a) Mode Misdiagnosis Rate



(b) Source Misdiagnosis Rate

Figure 10.10: Grouping approaches for KDE method

Chapter 11

Dynamic application of continuous evidence and ambiguous mode solutions

11.1 Introduction

Previous work [5] [4] addressed the topic of taking into account mode and evidence dependency when implementing Bayesian diagnosis techniques online. However, the dynamic techniques that were previously introduced were specific to unambiguous modes and discrete evidence. This chapter aims to tie up previous loose ends so that autodependent nature of modes and evidence can be taken into account.

When considering the material in this thesis, the existing autodependent mode procedure [4] is only affected by the introduction of ambiguous modes (continuous evidence will not affect the autodependent mode procedure). Likewise, introducing ambiguous modes will not affect the autodependent evidence procedure. Thus, this chapter proposes two solutions that can be applied independently:

1. Taking into account autodependent modes with ambiguous modes in history
2. Taking into account autodependent continuous evidence

11.2 Algorithm for autodependent modes

In this section, we consider the probability of each unambiguous mode at time t, and its propagation over time as illustrated in Figure 11.1. At time t, the probability of the mode set M is predicted from time $t-1$. The prediction is done via transformation by the probability transition matrix A. After prediction, it is updated by the evidence obtained at time t (denoted as E^t). Updating through evidence is done using Bayes' Rule. The predict-update procedure is applied recursively so that diagnosis is made by probability that is constantly being updated.

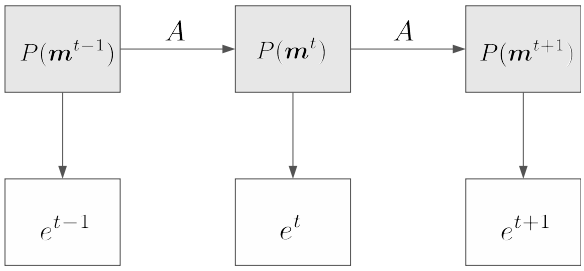


Figure 11.1: Mode autodependence

11.2.1 Probability transition matrix

The probability transition matrix A is used to predict the probabilities of each mode at the next time step.

$$\boldsymbol{p}(M^t) = A\boldsymbol{p}(M^{t-1})$$

where $\boldsymbol{p}(M^t)$ and $\boldsymbol{p}(M^{t-1})$ are column vectors of probability. The probability transition matrix A can be constructed as follows:

$$A = \begin{bmatrix} p(m_1^t|m_1^{t-1}) & p(m_1^t|m_2^{t-1}) & \cdots & p(m_1^t|m_n^{t-1}) \\ p(m_2^t|m_1^{t-1}) & p(m_2^t|m_2^{t-1}) & \cdots & p(m_2^t|m_n^{t-1}) \\ \vdots & \vdots & \ddots & \vdots \\ p(m_n^t|m_1^{t-1}) & p(m_n^t|m_2^{t-1}) & \cdots & p(m_n^t|m_n^{t-1}) \end{bmatrix}$$

where $p(m_i^t|m_j^{t-1})$ is the probability of mode i occurring given mode j occurring at the previous time instant. The transition probability matrix is constructed in such a manner that each column must sum to unity.

11.2.2 Review of second-order method

In Chapter 8, where the second-order Bayesion rule of combination was introduced, the likelihood was expressed in terms of the parameter set Θ.

$$p(E|M,\Theta) = \frac{S(E|M)n(M) + \sum\limits_{\boldsymbol{m}_k \supset M} \theta\{\frac{M}{\boldsymbol{m}_k}\} S(E|\boldsymbol{m}_k)n(\boldsymbol{m}_k)}{n(M) + \sum\limits_{\boldsymbol{m}_k \supset M} \theta\{\frac{M}{\boldsymbol{m}_k}\} n(\boldsymbol{m}_k)}$$

The second-order Bayesian method re-expressed the likelihood in terms of a second-order approximation

$$\Delta\Theta = \Theta - \hat{\Theta}$$

$$p(E|M,\Theta) = \hat{p}(E|M) + \boldsymbol{J}_L^m \Delta\Theta + \frac{1}{2}\Delta\Theta^T \boldsymbol{H}_L^m \Delta\Theta$$

where $\boldsymbol{J}_L^m$ is the likelihood Jacobian for mode M and $\boldsymbol{H}_L^m$ is the likelihood Hessian for mode M. In a similar manner, the prior probability was expressed as

$$p(M|\Theta) = \hat{p}(M) + \boldsymbol{J}_R^m \Delta\Theta + \frac{1}{2}\Delta\Theta^T \boldsymbol{H}_R^m \Delta\Theta$$

Here, if using prior probabilities from experts, it is common that priors do not have any ambiguity; in such cases, $\boldsymbol{J}_R^m$ and $\boldsymbol{H}_R^m$ are set to be zero matrices. By using the second-order rule of combination (introduced in Chpater 8), new Jacobians and Hessians can be calculated:

$$p(M|E,\Theta) = \hat{p}(M|E) + \boldsymbol{J}_P^m \Delta\Theta + \frac{1}{2}\Delta\Theta^T \boldsymbol{H}_P^m \Delta\Theta$$

where the second-order terms are calculated as

$$\begin{aligned}
\hat{p}(M|E) &= \frac{1}{K}\hat{p}(E|M)\hat{p}(M) \qquad (11.1)\\
\boldsymbol{J}_P^m &= \frac{1}{K}\left[\hat{p}(M)\boldsymbol{J}_L^m + \hat{p}(E|M)\boldsymbol{J}_R^m\right]\\
\boldsymbol{H}_P^m &= \frac{1}{K}\left[\hat{p}(M)\boldsymbol{H}_L^m + \hat{p}(E|M)\boldsymbol{H}_R^m + (\boldsymbol{J}_L^m)^T(\boldsymbol{J}_R^m) + (\boldsymbol{J}_R^m)^T(\boldsymbol{J}_L^m)\right]\\
K &= \sum_{k=1}^{n_m} \hat{p}(E|m_k)\hat{p}(m_k)
\end{aligned}$$

11.2.3 Second-order probability transition rule

By merit of the probability transition matrix, the probability of mode k can be expressed as

$$p(m_k^t|E^{t-1}) = \sum_{i=1}^{n} A(k,i)p(m_i^{t-1}|E^{t-1}) \qquad (11.2)$$

where, by applying this transformation, the posterior $p(m_i^{t-1}|E)$ at time $t-1$ is transformed into the prior $p(m_k^t)$ for time t. In this chapter we apply the rule in Eqn (11.1) to the second-order probability expressions developed in Chapter 8 to create a second-order probability transition rule. First, let us consider the parameterized probability

$$p(m_k^t|\Theta) = \hat{p}(M^k) + \boldsymbol{J}_R^{m_k} \Delta\Theta + \frac{1}{2}\Delta\Theta^T \boldsymbol{H}_R^{m_k} \Delta\Theta$$

By applying the probability transition rule in Eqn (11.2) to the parametrized probability above, one can collect terms, and obtain a new probability transition rule set for the reference probability, the Jacobian and Hessian respectively.

$$\hat{p}(m_k^t) = \sum_{i=1}^{n} A(k,i)\hat{p}(m_i^{t-1}|E) \qquad (11.3)$$

$$\boldsymbol{J}_R^{m_k} = \sum_{i=1}^{n} A(k,i)\boldsymbol{J}_R^{m_i} \qquad (11.4)$$

$$\boldsymbol{H}_R^{m_k} = \sum_{i=1}^{n} A(k,i)\boldsymbol{H}_R^{m_i} \qquad (11.5)$$

Thus, when transitioning to the next time step, the probability transition rules in Eqn (11.3), (11.4) and (11.5) are applied to each reference probability, Jacobian, and Hessian respectively in order to obtain the prior for the next time step. Updating can occur in the usual manner, i.e. the second-order Bayesian combination rule (however, keep in mind that Jacobians and Hessians for priors no longer tend to be zero matrices).

11.3 Algorithm

11.3.1 Algorithm for dynamic continuous evidence

The problem of autodependent discrete evidence has been dealt with previously by Qi and Huang [5]. The goal was to take into account evidence dependence as indicated in Figure 11.2. Here, the likelihood must contain previous evidence $p(E^t|E^{t-1}, M)$ which can be combined with prior probabilities $p(M)$ to obtain a posterior $p(M|E^t, E^{t-1})$

$$p(m_i|E^t, E^{t-1}) = \frac{p(E^t|E^{t-1}, m_i)p(m_i)}{\sum_k p(E^t|E^{t-1}, m_k)p(m_k)}$$

The key challenge is to estimate the likelihood $p(E^t|E^{t-1}, M)$.

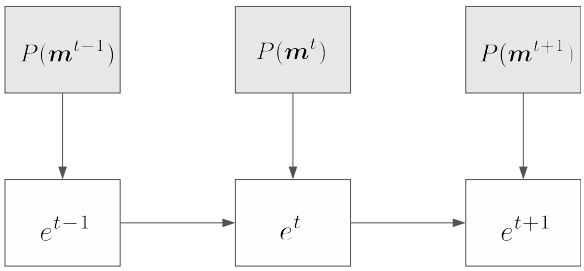


Figure 11.2: Evidence autodependence

Review of discrete evidence solution

In Chapter 2 it was previously shown that the likelihood for autodependent evidence can be estimated as

$$p(E^t|E^{t-1}, m_k) = \frac{n(E^t, E^{t-1}, m_k)}{n(E^{t-1}, m_k)} \tag{11.6}$$

where $n(E^t, E^{t-1}, m_k)$ is the number of times E^t, E^{t-1} and m_k jointly occur in the history, while $n(E^{t-1}, m_k)$ is the number of times $n(E^{t-1})$ and m_k jointly occur in the history. This solution also included the ability to use prior samples, and these prior samples can be simply combined with the historical data.

One of the challenges was that introducing autodependence increased the evidence space, as now we must condition on both E^t and E^{t-1}. Correlation ratio tests were proposed to determine if some dependencies could be neglected and thus narrowing the evidence space by independence assumptions. In this work, the *mutual information criterion* (MIC) is used as the independence testing method, as it functions for both discrete and continuous evidence.

Continuous evidence solution

One problem with the solution in Eqn (11.6) is that one cannot obtain $n(E^t \cap E^{t-1} \cap m_k)$ directly from continuous data. In Chapter 10, the kernel density estimate was used to approximate the likelihood

$$p(E|M) = \frac{n(E,M)}{n(M)}$$
$$p(E|M) \approx \frac{1}{N_K} \sum_{i=1}^{n} \frac{1}{|H|} K(H^{-1/2}e) = \hat{f}(E|M)$$

Thus, the conditioning over continuous evidence $n(E,M)$ was smoothed over using a kernel density estimate. In order to condition on both E^t and E^{t-1} the rule of conditioning is applied

$$p(Y|X) = \frac{p(X \cap Y)}{p(X)}$$

By applying the rule of conditioning to the kernel density estimation,

$$p(E^t|E^{t-1}, M^t) = \frac{p(E^t, E^{t-1}|M^t)}{p(E^{t-1}|M^t)} \tag{11.7}$$

From this result, one can see that two kernel density estimates are required:

1. The joint present and past evidence $p(E^t, E^{t-1}|M^t)$
2. The past evidence $p(E^{t-1}|M^t)$ (which is also equal to $p(E^t|M^t)$)

Thus, for dynamic evidence, one can estimate a kernel density function $p(E^t|M^t)$ as before, but an additional step is required: that the kernel density estimate of the present and past observations $p(E^t, E^{t-1}|M^t)$ also be estimated. To obtain the posterior, simply combine this new likelihood with a prior using Bayes' Theorem

$$p(E^t|E^{t-1}, m_i^t) = \frac{p(E^t, E^{t-1}|m_i^t)}{p(E^{t-1}|m_i^t)}$$
$$p(m_i^t|E^t, E^{t-1}) = \frac{p(E^t|E^{t-1}, m_i^t)p(m_i^t)}{\sum_k p(E^t|E^{t-1}, m_k^t)p(m_k^t)}$$

Dimensionality reduction

As was mentioned in Chapter 10, dimensionality can be an issue when dealing with a large number of instruments. However, dimensionality becomes an even more significant issue when taking dynamic evidence into account as $p(E^t|E^{t-1}, m_i^t)$ has twice the dimensionality as $p(E|m_i^t)$. Thus, dimensionality reduction techniques, such as grouping via mutual information criterion become an even more relevant practice.

11.3.2 Combining both solutions

The second-order probability transition rules in Eqn (11.3 - 11.5) and the continuous autodependent solution in Eqn (11.7) are complementary solutions that can be independently applied. When applying both solutions, the corresponding Bayesian network diagram takes the form shown in Figure 11.3.

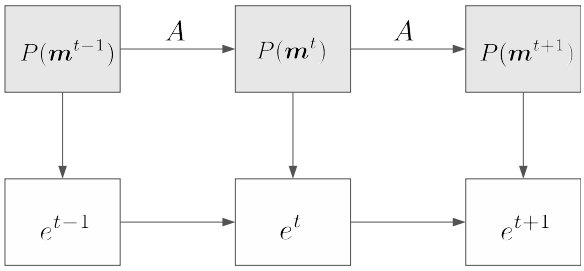


Figure 11.3: Evidence and mode autodependence

When applying both methods, one obtains the prior probability from the previous posterior according to the second-order transition rule:

$$\begin{aligned}
\hat{p}(m_k^t) &= \sum_{i=1}^{n} A(k,i)\hat{p}(m_i^{t-1}|E) \\
\boldsymbol{J}_R^{m_k} &= \sum_{i=1}^{n} A(k,i)\boldsymbol{J}_R^{m_i} \\
\boldsymbol{H}_R^{m_k} &= \sum_{i=1}^{n} A(k,i)\boldsymbol{H}_R^{m_i} \\
p(m_k^t|\Theta) &= \hat{p}(m_k^t) + \boldsymbol{J}_R^{m_k}\Delta\Theta + \frac{1}{2}\Delta\Theta^T\boldsymbol{H}_R^{m_k}\Delta\Theta
\end{aligned}$$

Then, the second-order terms $\hat{p}(E^t|E^{t-1}, m_k^t)$, $\boldsymbol{J}_L^{m_k}$ and $\boldsymbol{H}_L^{m_k}$ are calculated for the likeli-

hood expression

$$p(E^t|E^{t-1}, M^t, \Theta) = \frac{S(E|M^t)n(M^t) + \sum_{\boldsymbol{m}_k \supset M^t} \theta\{\frac{M^t}{\boldsymbol{m}_k}\} S(E^t|E^{t-1}, \boldsymbol{m}_k) n(\boldsymbol{m}_k)}{n(M^t) + \sum_{\boldsymbol{m}_k \supset M^t} \theta\{\frac{M^t}{\boldsymbol{m}_k}\} n(\boldsymbol{m}_k)}$$

where the support $S(E^t|E^{t-1}, \boldsymbol{m}_k)$ is calculated as

$$S(E^t|E^{t-1}, \boldsymbol{m}_k) = \frac{\hat{f}(E^t, E^{t-1}|\boldsymbol{m}_k)}{\hat{f}(E^t|\boldsymbol{m}_k)}$$

where $\boldsymbol{m}_k$ is a potentially ambiguous mode. The notation $\hat{f}(E^t, E^{t-1}|\boldsymbol{m}_k)$ indicates that the kernel density estimate uses evidence from times t and $t-1$ collected under the mode $\boldsymbol{m}_k$ (which can either be ambiguous or unambiguous). When values of $S(E^t|E^{t-1}, \boldsymbol{m}_k)$ are obtained, the terms $\hat{p}(E^t|E^{t-1}, m_k^t)$, $\boldsymbol{J}_L^{m_k}$ and $\boldsymbol{H}_L^{m_k}$ can be obtained by taking derivatives of the resulting expression for $p(E^t|E^{t-1}, M^t, \Theta)$.

Once the second-order terms $\hat{p}(E^t|E^{t-1}, m_k^t)$, $\boldsymbol{J}_L^{m_k}$ and $\boldsymbol{H}_L^{m_k}$ have been obtained, they are combined with the previously obtained $\hat{p}(m_k^t|E^t, E^{t-1})$ $\boldsymbol{J}_R^{m_k}$ and $\boldsymbol{H}_R^{m_k}$ terms using the second-order Bayesian combination rule.

11.3.3 Comments on usefulness

Mode autodependence

The effectiveness of taking account mode and evidence autodependence can vary quite significantly depending on the application. Taking autodependence of modes into account is particularly effective when modes change relatively slowly and if evidence is noisy. Noisy evidence can lead to relatively frequent false diagnosis results; however, if modes change relatively slowly, taking mode autodependence into account creates a time-weighted average on the diagnosis results. However, mode autodependence will create a diagnosis tool that is sluggish to respond when mode changes, as a strong prior would have been created from the previous mode over time.

If taking modes autodependence into account results in a system that is too sluggish, one can replace the transition matrix A with an exponent to the power of n which represents an acceleration factor.

$$A = A^n$$

For example, if $n = 2$, then it is assumed that the modes switch twice as frequently as previously thought. As n grows larger, the system responds more quickly to change. In fact, if $n \to \infty$, the resulting probability transition matrix will yield a flat prior for every time step which has the fastest possible reaction to a change in the process.

Evidence autodependence

Evidence autodependence tends to make the Bayesian diagnosis method slow to detect changes in the mode. In cases of strong evidence autodependence, the evidence trajectory follows a clear pattern in transition regions where values of evidence tend to slowly drift toward typical values for the new mode. This slow transition is often due to filtering or averaging done by the monitor. By documenting these transition regions, one now has a transition pattern which can be used to more quickly recognize changes in the operating mode; this is particularly beneficial if fast detection is desired. If one is more concerned about diagnosing slow-changing modes over longer-term periods, the benefits of accounting for evidence autodependence are much less significant.

In summary, taking account autodependence in the evidence is most effective for systems that have strongly autodependent (or slow-responding) evidence and frequently changing modes. Primarily, this is because of the time-sensitive nature for diagnosing systems with rapidly changing modes. Furthermore, taking autodependence into account is most manageable when the possible operating modes are few, as it requires data to be collected from transition regions between all modes.

11.4 Illustrative Example of Proposed Methodology

11.4.1 Introduction

Again, for the tutorial, we consider the control loop system shown in Figure 11.4.

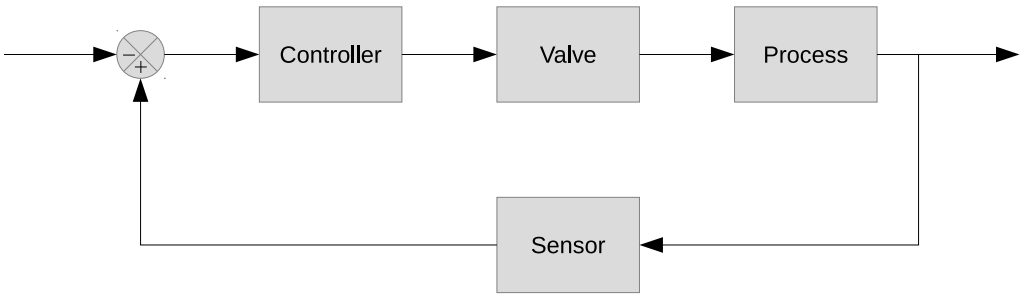


Figure 11.4: Typical control loop

11.4.2 Offline Step 1: Historical data collection

The first step is to go through the historical data and note the instances where each of the four possible modes occurs, and collect data belonging to the mode.

1. m_1 $[0, 0]$ where bias and stiction do not occur

2. m_2 $[0, 1]$ where bias does not occur but stiction does

3. m_3 $[1, 0]$ where bias occurs but stiction does not

4. m_4 $[1, 1]$ where both bias and stiction occur

If ambiguous modes exist in the data, one must also collect data according to the ambiguous modes that appear in the history. For this system, the possible ambiguous modes are

1. $\{m_1, m_3\}$ $[\times, 0]$ where bias is undetermined and stiction does not occur

2. $\{m_2, m_4\}$ $[\times, 1]$ where bias is undetermined and stiction occurs

3. $\{m_1, m_2\}$ $[0, \times]$ where bias does not occur and stiction is undetermined

4. $\{m_3, m_4\}$ $[1, \times]$ where bias occurs and stiction is undetermined

5. $\{m_1, m_2, m_3, m_4\}$ $[\times, \times]$ where both bias and stiction are undetermined

Because the methods deal with kernel density estimation and other computationally intensive techniques, the application will be given in terms of MATLAB code.

11.4.3 Offline Step 2: create temporal data

Let us consider the MATLAB cell variable `Data` which has all data collected into modes, both unambiguous and ambiguous. We can construct a new data set that includes the data from the previous time step (called the one-step temporal data). This is done by copying the data, deleting the first row of one set, and deleting the last row of the second set, then combining the two data sets.

```
for m = 1:length{Data}
    e1 = Data{m};
    e2 = e1;
    e1(end,:) = []; %Delete last row of previous evidence
    e2(1,:) = [];   %Delete first row of current evidence
    DataT{m} = [e1,e2];
end
```

11.4.4 Offline Step 3: Mutual Information Criterion (optional, but recommended)

As done in the kernel density estimation procedure, the mutual information criterion is used to group evidence into independent groups. However, grouping must now be done for two sets of data, the original data `Data` and the temporal data `DataT`.

```
    MICmatrix = zeros(length{Data{m});
    ne = length(Data{m}(1,:)); %number of evidence sources
    for j = 1:ne
```

```
        for i = (j+1):ne
            MICmatrix(i,j) = fMIC(Data{m}(:,i,j))
        end
    end

    MICmatrixT = zeros(length{DataT{m});
    ne = length(DataT{m}(1,:)); %number of evidence sources
    for j = 1:ne
        for i = (j+1):ne
            MICmatrixT(i,j) = fMIC(DataT{m}(:,i,j))
        end
    end
```

where `fMIC` is the function described in Chapter 10. Using the mutual information criterion matrices, the original data, and the one-step temporal data can be grouped into roughly independent groups `Groups` for original data, and `GroupsT` for one-step temporal. The MATLAB cell variables `Groups` and `GroupsT` contain cell arrays `groups` for each mode. The cell array `groups` pertains to a specific mode and contains vectors that consist of grouped instruments; each vector represents a group.

If one does not wish to break the evidence down into independent groups, it is still recommendable to evaluate the MIC in order to evaluate autodependence. For autodependence, the MIC is evaluated for two data sets: x_0 which is the original data set, and x_1 which is the same as x_0 except that it is shifted by one time sample. When evaluating the MIC for kernel densities, if the MIC is less than 0.1, autodependence can be ignored.

Offline Step 4: Kernel Density bandwidths for original and one-step temporal data

Kernel density estimation is done in the same manner as done in Chapter 10, where bandwidth and data are stored in `KDE`. However, now ambiguous modes also need to be estimated, and kernel density estimates now exist for both original and one-step temporal data.

```
    for M = 1:length{Data}
        %Kernel Density Estimate for original data
        groups = Groups{M};
        for g = 1:length{groups}
            KDE{M}(g) = fKernelEstimateNorm(Data{M}(:,groups{g}));
        end

       %Kernel Density Estimate for one-step temporal data
       groups = GroupsT{M};
        for g = 1:length{groups}
            KDE{M}(g) = fKernelEstimateNorm(DataT{M}(:,groups{g}));
        end
    end
```

11.4.5 Offline Step 5: Calculate reference values

Again, we make use of the parameter matrix $\hat{\mathbf{\Theta}}$ in the same manner as the second-order Bayesian method.

$$\hat{\mathbf{\Theta}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hat{\theta}\{\frac{m_1}{m_1,m_2}\} & \hat{\theta}\{\frac{m_2}{m_1,m_2}\} & 0 & 0 \\ \hat{\theta}\{\frac{m_1}{m_1,m_3}\} & 0 & \hat{\theta}\{\frac{m_3}{m_1,m_3}\} & 0 \\ 0 & \hat{\theta}\{\frac{m_2}{m_2,m_4}\} & 0 & \hat{\theta}\{\frac{m_4}{m_2,m_4}\} \\ 0 & 0 & \hat{\theta}\{\frac{m_3}{m_3,m_4}\} & \hat{\theta}\{\frac{m_4}{m_3,m_4}\} \\ \hat{\theta}\{\frac{m_4}{m_1,m_2,m_3,m_4}\} & \hat{\theta}\{\frac{m_4}{m_1,m_2,m_3,m_4}\} & \hat{\theta}\{\frac{m_4}{m_1,m_2,m_3,m_4}\} & \hat{\theta}\{\frac{m_4}{m_1,m_2,m_3,m_4}\} \end{bmatrix}$$

11.4.6 Online Step 1: Obtain prior second-order terms

One obtains prior second-order terms for this step, from the posterior second-order terms in the previous step.

```
%number of historical modes and unambiguous modes
[nM,nm] = size(ThetaHat);

for m = 1:nm
    Prior(m).probability = 0;
    Prior(m).jacobian = zeros(1,(nA*nm));
    Prior(m).hessian = zeros((nA*nm),(nA*nm));

    for k = 1:nm
        Prior(m).probability = Prior(m).probability + Posterior(m).probability*A(m,k);
        Prior(m).jacobian = Prior(m).jacobian + Posterior(k).jacobian*A(m,k);
        Prior(m).hessian = Prior(m).hessian + Posterior(k).hessian*A(m,k);
    end
end
```

11.4.7 Online Step 2: Calculate support

When a new observation `e` has been made, one can calculate support for each mode. If multiple independent groups are present, the kernel density estimates can be multiplied together

```
%we have observed current evidence e1
%we collected previous evidence e0
eT = [e0,e1];
S = ones(length(Data),1);
for M = 1:length(Data)
    groups = Groups{M};
    for g = 1:length(groups)
        %Numberator and Denominator P(e1|e0,m)=P(e0,e1|m)/P(e0|m)
        PN = fKernelDensity(eT(groups{g}),KDET{M}(g));
        PD = fKernelDensity(e0(groups{g}),KDE{M}(g));
```

```
11
12          %If groups are independent, estimate support by multiplying
13          S(M) = S(M)*(PN/PD);
14      end
15  end
```

11.4.8 Online Step 3: Calculate second-order terms

Using the calculated support values (S), one can obtain the second-order terms as was previously done in Chapter 10.

```
1   function Lik = O2Terms(S,N,ThetaHat)
2       [nM,nm] = size(ThetaHat);
3       nA = nM - nm;
4
5       %Theta parameters for ambiguous modes
6       ThetaA = ThetaHat( (nm+1):end,:);
7
8       for m = 1:nm
9           %Initialize second-order terms
10          Lik(m).probability = 0;
11          Lik(m).jacobian = zeros(1,(nA*nm));
12          Lik(m).hessian = zeros((nA*nm),(nA*nm));
13
14          %Obtain numerators and denominators for the likelihood
15          SN = S.*N;  %Multiplication of S and N, useful for later
16          Num = SN'*ThetaHat(:,m);
17          Den = N'*ThetaHat(:,m);
18
19          Jac = zeros(1,nA);
20          Hes = zeros(nA,nA);
21
22          %Find indices where Theta values are not forced to be zero
23          Ind = find((ThetaA(:,m)~=0))';
24          for i = Ind
25             Jac(1,i) = SN(i)/Den - Num*N(i)/(Den^2);
26          end
27
28          for i = Ind
29              for j = Ind
30                  Hes(i,j) = -(N(i)*SN(j)+N(j)*SN(i))/(Den^2) + 2*Num*N(i)*N(j)/(Den^3);
31              end
32          end
33
34          %Find indices relevant to the current mode
35          ActiveInd = ((m-1)*nA+1):(m*nA);
36
37          Lik(m).probability = Num/Den;
38          Lik(m).jacobian(ActiveInd) = Jac;
39          Lik(m).hessian(ActiveInd,ActiveInd) = Hes;
40
41      end
```

11.4.9 Online Step 4: Combining prior and likelihood terms

After obtaining the prior and likelihood probabilities, we can perform the second-order rule of combination in order to obtain the posterior second-order terms.

```
function Post = SecondOrderComb(Prior,Lik)
nm = length(Prior); %number of modes
%Normalization Constant
for m = 1:nm
    K = K + Prior(m).probability * Lik(m).probability;
end

for m = 1:nm
    PP = Prior(m).probability;
    PL = Lik(m).probability;
    JP = Prior(m).jacobian;
    JL = Lik(m).jacobian;
    HP = Prior(m).hessian;
    HL = Lik(m).hessian;

    Post(m).probability = 1/K*(PP*PL);
    Post(m).jacobian = 1/K*(JP*PL + JL*PP);
    Post(m).hessian = 1/K*(HP*PL + HL*PP + JL'*JP + JP'*JL);
end
```

The diagnosis can be made by selecting the posterior likelihood `Post(m).probability` which has the maximum value (the point estimate method), or one can also use the expected value method which makes use of the second-order terms to calculate an expected value, as mentioned in Chapter 10.

11.5 Simulated Case

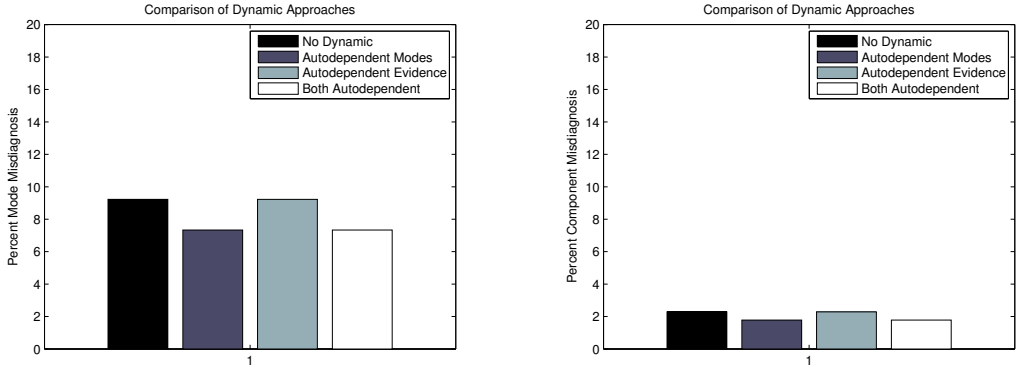
To create autodependent modes during the simulation, the mode was switched at random according to a given switching probability. Autodependence in evidence however, is a function of the monitors and cannot be easily created. Nevertheless, evidence autodependence can be easily reduced by sampling monitor data at a slower rate. Thus, highest amounts of autodependence in the evidence are present when monitor data is sampled at its native frequency.

Four methods of interest were tested on the Tennessee Eastman simulation. The first method did not use any autodependent techniques, the second method only took into account autodependent modes, the third method only took into account autodependent evidence, while the final method took into account autodependence in both modes and evidence. When the dynamic evidence was considered, the MIC was evaluated in order to determine if autodependence was strong enough to be considered.

Results for the four techniques can be seen in Figure 11.5. For the Tenessee-Eastman simulation, it was found that autodependence in evidence was quite weak, as can be seen in the results. However, even before the results were obtained, it was noticed that the MIC values for autodependence averaged at around 0.05, which is a relatively weak value for autodependence (recall that if the MIC was less than 0.1, autodependence was ignored). The low autodependence in evidence was probably caused from a slow sampling time.

In contrast to mode autodependence however, modes tended to change relatively slowly, thus modes tended to have strong autodependence in the simulation. As is seen in Figure

11.5, taking mode autodependence into account had significant performance improvement.

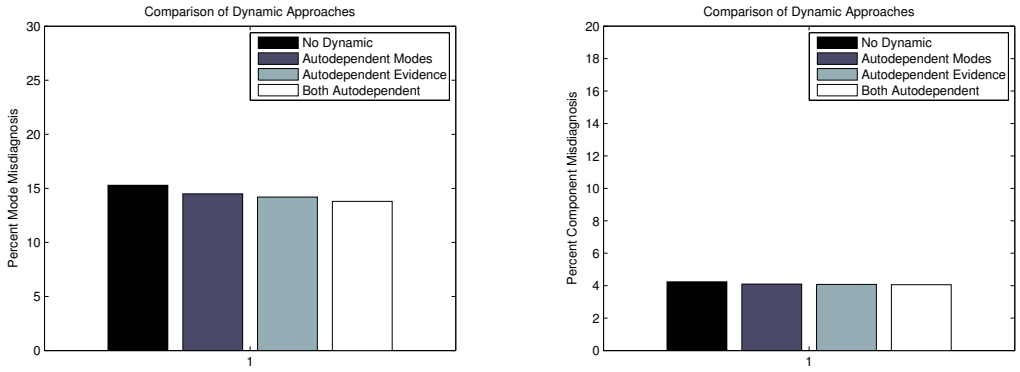


(a) Mode Misdiagnosis Rate (b) Source Misdiagnosis Rate

Figure 11.5: Comparison of dynamic methods

11.6 Bench Scale Case

The bench scale system exhibited fairly different behaviour from the simulated system. In the bench scale system, the modes were set to switch relatively quickly and data was sampled at a slower rate. MIC values for autodependence tended toward values of 0.08 which is closer to the threshold where autodependence is taken into account. From Figure 11.6, one can see that taking modes and evidence autodependence into account has a modest effect on the performance.

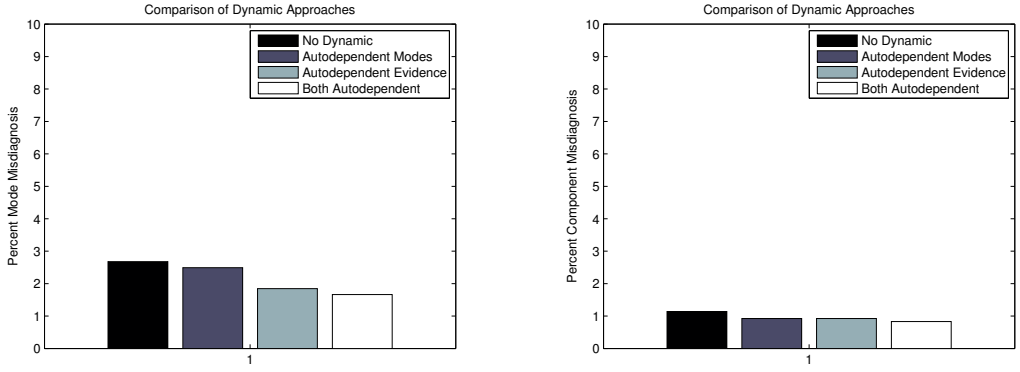


(a) Mode Misdiagnosis Rate (b) Source Misdiagnosis Rate

Figure 11.6: Comparison of dynamic methods

11.7 Industrial Scale Case

For the industrial system, data was sampled relatively quickly, but only small groups of data were selected as modes changed extremely infrequently. In order to better assess conditions where modes were switching, validation data for each mode was divided into groups, and these groups were shuffled so that it would appear that modes switched more frequently. Results can be seen in Figure 11.7. In this case, one can see that accounting for autodependence in the evidence had a stronger effect than accounting for autodependence in the modes. The efficacy of accounting for evidence autodependence is mainly due to the fact that autodependence was relatively strong (MIC values averaged at 0.2) and that modes switched relatively quickly.



(a) Mode Misdiagnosis Rate

(b) Source Misdiagnosis Rate

Figure 11.7: Comparison of dynamic methods

Chapter 12

Concluding remarks and recommendations for future work

12.1 Concluding Remarks

12.1.1 Summary of proposed solutions

The principal objectives of this thesis were to further develop the work in a data-driven Bayesian approach to detect and diagnose problematic components in process systems. Topics related to enhancing this approach include ambiguous modes, Dempster-Shafer theory, kernel density estimation and bootstrapping, all of which had their fundamentals explained in the *tutorial chapter*. Mores specifically, the main contributions of this paper can be summarized as follows:

- A parametrization sheme (respect to probability parameters Θ) was developed in order to formulate the likelihood estimation problem to account for ambiguous modes. Additionally, in order to perform Bayesian inference, a second-order approximation (with respect to Θ) of Bayes' Theorem was developed.
- The Dempster-Shafer Theory solution was derived using the Θ parametrization and it was shown that Dempster-Shafer Theory was relevant for estimating direct probabilities, but was not relevant for estimating likelihoods. A modification of Demspter-Shafer Theory was proposed in order to express the Basic Belief Assignment (BBA) to better-fit the problem at hand. Demspter's Rule of Combination was also modified so that it can be applied to the new generalized form of the BBA.
- Kernel density estimation was proposed as a solution to implement continuous evidence in our Bayesian diagnosis solution. It was shown through examples and through proofs that continuous methods perform as well or better than discrete methods. Solutions to missing the evidence problem and dimensionality issues were also discussed.
- Bootstrapping and component-wise diagnosis were proposed as methods to deal with spares modes within the process data. A modification over the previous bootstrap-

ping approach (called smoothed bootstrapping which is related to kernel density estimation) was proposed in order to achieve better sampling properties. While the Bootstrapping technique requires process knowledge, adopting component-wise diagnosis can be readily implemented in any framework. It also enjoys the advantage of reducing dimensionality, an issue discussed in Chapter 6.

- Implementing kernel density estimation and ambiguous modes will affect the previously proposed solutions for auto-dependent data and auto-dependent modes respectively. Chapter 11 extends the auto-dependent data and mode concepts presented in earlier work in order to enable their application toward ambiguous modes and continuous evidence.

12.1.2 Unified Bayesian framework

Each of the proposed methods can be applied simultaneously with the others (with exception of the second-order Bayesian method and the Generalized Dempster-Shafer method, the user must chose one of these two techniques); in this way, each of these solutions can be fit to a final unified framework, implemented as follows:

Offline Step 1: Break the system down into components

This step is mentioned in Chapter 7 where we break the system down into smaller components in order to reduce the mode dimension. In this way, the mode is described as a vector of P component states $M = [C^1, C^2, \ldots, C^P]$. After the system is broken down, it is also recommended that one tests the sensitivity of the monitors for each mode. Removing insensitive monitors reduces dimensionality and improves robustness of the diagnosis method.

Offline Step 2: Fill in data from missing modes

If there are any missing modes, and if there is adequate information about the system, it is possible to simulate data for extra modes using the bootstrap method mentioned in Chapter 7. Breaking down the system into components however, will reduce the likelihood that important modes will be missing.

Offline Step 3: Perform grouping for each component

In this optional step, the user can employ the Mutual Information Criterion to locate independent groups of monitors, and further reduce the dimension of the data. Breaking down the system into components and selecting only sensitive monitors however, will result in an already reduced dimension; grouping should only be considered if the dimension is large (e.g. greater than 5).

Offline Step 4: Identify any ambiguous modes and determine the size of Θ

In this step, one should identify any modes that are ambiguous. One may consider discarding ambiguous modes if the amount of data pertaining to these modes is small (e.g. less than 10%). Note that breaking down the system into smaller components can reduce the proportion of data belonging to such modes; furthermore, each component will have its own Θ parameter set. Using these ambiguous modes, one should determine the size of Θ.

Offline Step 5: Construct prior probabilities

Prior probabilities of unambiguous modes (or component sates) should be constructed; if taking ambiguous modes into account, the Jacobian and Hessian matrices of the prior's second-order approximation should also be constructed (they can be zero matrices).

Offline Step 6: Construct kernel density estimates

If continuous data is available, it is highly recommended that kernel density estimates be used over discrete estimates. The kernel density estimate

$$p(E|C) \approx f(E|C)$$

should be constructed according to Chapter 6 for all component modes in C (including ambiguous modes if present). In addition, if one wishes to take into account autodependent evidence, the additional kernel density estimate

$$p(E^t, E^{t-1}|C) \approx f(E^t, E^{t-1}|C)$$

should be obtained so that the conditional likelihood can be estimated as

$$p(E^t|E^{t-1}C) \approx \frac{f(E^t, E^{t-1}|C)}{f(E^{t-1}|C)}$$

Note that the kernel density estimate $f(E^{t-1}|C)$ is the same as $f(E^t|C)$, the only difference is that E^{t-1} is used as input instead of E^t.

Offline Step 7: Fill in missing historical evidence

If there are any missing monitor values, one can use the kernel density regression technique to fill in missing data.

Offline Step 8: Define probability transition matrix

If one is considering the effect of autodependent modes, a probability transition matrix should be constructed. This matrix is essentially a tuning parameter, but one can use data to estimate the switching probability of modes or component states.

Online Step 1: Probability transition

Using the posterior estimate from the previous step (or an initial posterior) use switching probabilities to obtain the prior probabilities for the current time. If considering ambiguous modes, the update rule take the following form:

$$\hat{p}(C^t|E^{t-1}) = \sum_{i=1}^{n} p(C^t|c_i^{t-1})\hat{p}(C^{t-1}|E^{t-1})$$
$$\boldsymbol{J}_{(C^t|E^{t-1})} = \sum_{i=1}^{n} p(C^t|c_i^{t-1})\boldsymbol{J}_{(C^{t-1}|E^{t-1})}$$
$$\boldsymbol{H}_{(C^t|E^{t-1})} = \sum_{i=1}^{n} p(C^t|c_i^{t-1})\boldsymbol{H}_{(C^{t-1}|E^{t-1})}$$

Online Step 2: Kernel density estimation

At each time step, new evidence E^t is obtained. Using this evidence, one can construct probabilities from kernel density estimates. When considering autodependent evidence, the support function is

$$S(E^t|E^{t-1}, \boldsymbol{C}) \approx \frac{f(E^t, E^{t-1}|\boldsymbol{C})}{f(E^{t-1}|\boldsymbol{C})}$$

Otherwise, a simple kernel density estimate can be used

$$S(E^t|\boldsymbol{C}) \approx f(E^t|\boldsymbol{C})$$

Online Step 3: Define likelihood function

If ambiguous modes (or equivalently, component states) are present in the data, one has to define the likelihood as a function of Θ

$$p(E|E^{t-1}, C, \Theta) = \frac{\sum\limits_{\boldsymbol{C}\supseteq C} \theta\{\frac{C}{\boldsymbol{C}}\} S(E^t|E^{t-1}, \boldsymbol{C}) n(\boldsymbol{C})}{\sum\limits_{\boldsymbol{C}\supseteq C} \theta\{\frac{C}{\boldsymbol{C}}\} n(\boldsymbol{C})}$$

This expression must then be converted to the second-order approximation format

$$p(E|E^{t-1}, C, \Theta) \approx \hat{p}(E|E^{t-1}, C) + \boldsymbol{J}_{(E|E^{t-1},C)}\Delta\Theta + \frac{1}{2}\Delta\Theta^T \boldsymbol{H}_{(E|E^{t-1},C)}\Delta\Theta$$

If there are no ambiguous modes, the likelihood will simply be

$$p(E|E^{t-1}, C) = S(E^t|E^{t-1}, \boldsymbol{C})$$

Online Step 4: Combine likelihood functions

If the evidence is separated into independent groups, one must use the second-order combination rule to obtain a combined likelihood estimate. For example, if evidence is separated into two independent groups $E = [E_1, E_2]$ one can obtain the combined second-order likelihood as

$$\begin{aligned}
\hat{p}(E|C) &= \hat{p}(E_1|C)\hat{p}(E_2|C) \\
\boldsymbol{J}_{(E|C)} &= \left[\boldsymbol{J}_{(E_1|C)}\hat{p}(E_2|C) + \boldsymbol{J}_{(C|E_1)}\hat{p}(C|E_2)\right] \\
\boldsymbol{H}_{(E|C)} &= \left[\boldsymbol{H}_{(E_1|C)}\hat{p}(E_2|C) + \boldsymbol{H}_{(C|E_1)}\hat{p}(C|E_2) + \right. \\
&\qquad \left. \boldsymbol{J}^T_{(C|E_1)}\boldsymbol{J}_{(C|E_2)} + \boldsymbol{J}^T_{(C|E_1)}\boldsymbol{J}_{(C|E_2)}\right]
\end{aligned}$$

Online Step 5: Combine likelihood and prior functions

Once the likelihood of all evidence is obtained, the likelihood can be combined with the prior (obtained in Online Step 1).

$$\begin{aligned}
\hat{p}(C^t|E^t) &= \frac{1}{p(E^t|E^{t-1})}\hat{p}(E^t|E^{t-1}, C)\hat{p}(C^t|E^{t-1}) \\
\boldsymbol{J}_{(C^t|E^t)} &= \frac{1}{p(E^t|E^{t-1})}\left[\boldsymbol{J}_{(C^t|E^{t-1})}\hat{p}(E^t|E^{t-1}, C) + \boldsymbol{J}_{(E^t|E^{t-1},C)}\hat{p}(C^t|E^{t-1})\right] \\
\boldsymbol{H}_{(C^t|E^t)} &= \frac{1}{p(E^t|E^{t-1})}\left[\boldsymbol{H}_{(C^t|E^{t-1})}\hat{p}(E^t|E^{t-1}, C) + \boldsymbol{H}_{(E^t|E^{t-1},C)}\hat{p}(C^t|E^{t-1}) + \right. \\
&\qquad \left. \boldsymbol{J}^T_{(C^t|E^{t-1})}\boldsymbol{J}_{(E^t|E^{t-1},C)} + \boldsymbol{J}^T_{(E^t|E^{t-1},C)}\boldsymbol{J}_{(C^t|E^{t-1})}\right]
\end{aligned}$$

Online Step 6: Diagnose components and modes

From the component posteriors, one can diagnose each component separately (which would define a mode) or construct mode probabilities using the product rule

$$p(M^t|E^t) = \prod_P p(C^t_P|E^t)$$

If $p(C^t_P|E^t)$ is a second-order function with respect to Θ, one must use the second-order combination rule to obtain the second-order product. To diagnose a mode from the second-order product, reference probabilities $\hat{p}(M^t|E^t)$ or the expected values $E_\Theta[p(M^t|E^t, \Theta)f(\Theta)]$ can be used to obtain a diagnosis.

12.1.3 Summary of application cases

Three test-bed systems were also introduced in order to evaluate the performance of each of these techniques.

1. The well-known Tennessee-Eastman simulation simulated a chemical process, and includes a built-in set of faults. This system was used to evaluate

- The second-order Bayesian approach for ambiguous modes (Chapter 8)
- The Generalized Dempster-Shafer approach for ambiguous modes (Chapter 9)
- The kernel density estimation approach for continuous data (Chapter 10)
- Dynamic adaptations of the second-order Bayesian approach and the continuous evidence approach (Chapter 11)

2. A bench-scale hybrid tank system exists among the experimental systems owned by Dr. Huang's research group. This system has lines between the tanks that can be opened to produce leaks. This system was used to evaluate

   - The Bootstrapping and component-based diagnosis approaches to address sparse modes (Chapter 7)
   - The second-order Bayesian approach for ambiguous modes (Chapter 8)
   - The Generalized Dempster-Shafer approach for ambiguous modes (Chapter 9)
   - The kernel density estimation approach for continuous data (Chapter 10)
   - Dynamic adaptations of the second-order Bayesian approach and the continuous evidence approach (Chapter 11)

3. Data from a solids handling facility in the Canadian Athabasca Oil Sands was used as a final test-bed solution. This system has been operating under a number of modes that are typically given by a scheduling variable. This system was used to evaluate

   - The second-order Bayesian approach for ambiguous modes (Chapter 8)
   - The Generalized Dempster-Shafer approach for ambiguous modes (Chapter 9)
   - The kernel density estimation approach for continuous data (Chapter 10)
   - Dynamic adaptations of the second-order Bayesian approach and the continuous evidence approach (Chapter 11)

12.2 Recommendations for Future Work

The following topics may be worthy of future investigations:

- The current implementation of Bayesian diagnosis is strictly a supervised learning problem. However, kernel density estimation provides a powerful framework for clustering which can be used as a tool to help perform the separation of data into modes (a task that must be performed before this Bayesian framework can be used).

- Unsupervised learning through kernel density estimation may also be potentially used to address the ambiguous mode problem. Data from ambiguous modes can potentially be sorted out into their respective specific modes by a combination of kernel density clustering and the use of specific mode reference data.

- Further opportunities in unsupervised learning can also be used to help reduce data dimensionality. Work by Gonzalez et al. [29] showed that Bayesian networks and kernel density estimation complement each other quite well to decompose high-dimensional non-linear distributions into a more manageable form. Unsupervised learning of Bayesian networks through kernel density estimation would be a very interesting (and potentially challenging problem) worthy of investigation.

- In this work, two bandwidth selection techniques were proposed for kernel density estimation: the optimal Gaussian bandwidth and an commonly-used adaptive kernel technique. In a related paper [54], an advanced bandwidth selection method was tested with disappointing results. Bandwidth selection is an active area or research in kernel density estimation, and several authors [62] [73] rigorously describe the performance metrics of bandwidth selectors. It may be advantageous to consider other bandwidth selection techniques.

- Diagnosing a mode will prompt the user to either maintain current operation or take action to rectify the situation. Taking action often has economic consequences. For practical implementation, it is important to consider the economic implications of each decision made so that the recommended actions are economically optimal.

Bibliography

[1] F. Qi, B. Huang, A Bayesian Approach for Control Loop Diagnosis with Missing Data, AIChE Journal 56 (1) (2010) 179–195.

[2] A. Pernestal, A Bayesian Approach to Fault Isolation with Application to Diesel Engines, Ph.D. thesis, KTH School of Electrical Engineering, 2007.

[3] F. Qi, Bayesian Approach for Control Loop Diagnosis, Ph.D. thesis, University of Alberta, 2011.

[4] F. Qi, B. Huang, Dynamic Bayesian Approach for Control Loop Diagnosis with Underlying Mode Dependency, AIChE Journal 49 (2010) 8613–8623.

[5] F. Qi, B. Huang, Bayesian methods for control loop diagnosis in the presence of temporal dependent evidences, Automatica 47 (2011) 1349–1356.

[6] I. Nimmo, Adequately address abnormal situation operations., Chemical Engineering Progress 91 (9) (2003) 36–45.

[7] V. Venkatasubramanian, R. Rengaswamy, K. Yin, S. N. Kavuri, A review of process fault detection and diagnosis Part I: Quantitative model-based methods, Computers and Chemical Engineering 27 (2003) 293–311.

[8] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, A review of process fault detection and diagnosis Part II: Qualitative models and search strategies, Computers and Chemical Engineering 27 (2003) 313–326.

[9] V. Venkatasubramanian, R. Rengaswamy, K. Yin, S. N. Kavuri, A review of process fault detection and diagnosis Part III: Process history based methods, Computers and Chemical Engineering 27 (2003) 327–346.

[10] U. N. Lerner, Hybrid Bayesian networks for reasoning about complex systems, Ph.D. thesis, Stanford University, 2002.

[11] C. Romessis, K. Mathioudakis, Bayesian network approach for gas path fault diagnosis, Journal of Engineering for Gas Turbines and Power 128 (2006) 64–72.

[12] P. Frank, Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-based Redundancy A Survey and Some New Results*, Automatica 26 (3) (1990) 459–474.

[13] M. Desai, A. Ray, A fault detection and isolation methodology, in: Proc. 20th Conf. on Decision and Control, 1363–1369, 1981.

[14] H. Jones, Failure detection in linear systems, Ph.D. thesis, MIT, Cambridge, MA, 1973.

[15] R. N. Clark, D. C. Fosth, V. M. Walton, Detection instrument malfunctions in control systems, IEEE Trans. Aerospace Electron. Syst AES-II (1975) 465–473.

[16] A. Willsky, Detection instrument malfunctions in control systems, Automatica 12 (1976) 601–611.

[17] E. Garcia, P. Frank, Deterministic nonlinear observer-based approaches to fault diagnosis: a survey, Control Engineering Practice 5 (5) (1997) 663–670.

[18] A. Hagenblad, F. Gustafsson, I. Klein, A comparison of two methods for stochstic fault detection: the parity space approach and principle component analysis, in: IFAC Symposium System Identification., 2004.

[19] P. Frank, Analytical and Qualitative Model-based Fault Diagnosis – A Survey and Some New Results, European Journal of Control 2 (1996) 6–28.

[20] R. Isermann, B. Freyermuth, Process fault diagnosis based on process model knowledge, Parts I (Principles) and II (Case study experiments), ASME Journal of Dynamic Systems, Measurement Control 113 (1991) 620–633.

[21] R. Isermann, Fault diagnosis of machines via parameter estimation and knowledge processing, Automatica 29 (1993) 815–836.

[22] A. Rault, D. Jaume, M. Verge, Industrial fault detection and localization, in: IFAC 9th workd congress, Budapest, vol. 4, 1789–1792, 1984.

[23] R. Gonzalez, B. Huang, F. Xu, A. Espejo, Dynamic bayesian approach to gross error detection and compensation with application toward an oil sands process, Chemical Engineering Science 67 (1) (2012) 44–56.

[24] U. Lerner, R.Parr, D. Koller, G. Biswas, Bayesian Fault Detection and Diagnosis in Dynamic Systems, in: AAAI-00 Proceedings, 2000.

[25] Z. Ge, Z. Song, Mixture Bayesian Regularization Method of PPCA for Multimode Process Monitoring, Process Systems Engineering 56 (11) (2010) 2838–2849.

[26] M. Tipping, C. Bishop, Probabilistic principal component analysis, Journal of the Royal Statistical Society 61 (3) (1998) 611–622.

[27] M. Tipping, C. Bishop, Mixtures of Propabilistic Principal Component Analysers, Neural Computation 11 (2) (1999) 443–482.

[28] J.-M. Lee, I.-B. Lee, S. Qin, Fault detection and diagnosis based on modified independent component analysis, AIChE Journal 52 (2006) 3501–3514.

[29] R. Gonzalez, B. Huang, E. Lau, Process monitoring using kernel density estimation and Bayesian networking with an industrial case study, submitted to ISA Transactions, 2014.

[30] T. Bayes, An essay towards solving a problem in the doctrine of chances, Biometrika 45 (1764/1958) 296–315.

[31] T. Harris, Assessment of Control Loop Performance, Canadian Journal of Chemical Engineering 67 (1989) 856–861.

[32] B. Huang, S. Shah, K. Kwok, On-line control performance monitoring of mimo processes, in: American Control Conference, 1995.

[33] T. Harris, F. Boudreau, J. F. MacGregor, Performance assessment using of multivariable feedback controllers, Automatica 32 (1996) 1505–1518.

[34] B. Huang, S. Shah, Performance Assessment of Control Loops, Springer, 1999.

[35] S. L. Shah, R. Patwardhan, B. Huang, Multivariate controller performance analysis: Methods, applications and challenges, in: Chemical Process Control Conference, 2001.

[36] J. Gao, R. Patwardhan, K. Akamatsu, Y. Hashimoto, G. Emoto, S. L.Shah, Performance evaluation of two industrial MPC controllers, Control Engineering Practice 11 (2003) 1371–1387.

[37] J. Schafer, A. Cinar, Multivariable MPC system performance assessment, monitoring, and diagnosis, Journal of Process Control 14 (2004) 113–129.

[38] M. Jelali, B. Huang, Detection and Diagnosis of Stiction in Control Loops: State of the Art and Advanced Methods, Springer London, 2009.

[39] R. Srinivasan, R. Rengaswamy, S. Narasimhan, R. Miller, A curve fitting method for detecting valve stiction in oscillation control loops, Industrial Engineering Chemistry Research 44 (2005) 6719–6728.

[40] Q. P. He, J. Wang, M. Pottmann, S. J. Qin, A curve fitting method for detecting valve stiction in oscillating control loops, Ind. Eng. Chem. Res. 46 (2007) 4549–4560.

[41] M. Choudhury, M. Jain, S. Shah, Stiction – definition, modelling, detection and quantification, Journal of Process Control 18 (2008) 232–243.

[42] B. Huang, On-line closed-loop model validation and detection of abrupt parameter changes, Journal of Process Control 11 (2007) 699–715.

[43] H. Jiang, B. Huang, S. Shah, Closed-loop model validation based on the two-model divergence method, Journal of Process Control 19 (2009) 644–655.

[44] A. Badwe, R. Gudi, R. Patwardhan, S. Shah, S. Patwardhan, Detection of model-plant mismatch in MPC applications, Journal of Process Control 19 (2009) 1305–1313.

[45] R. S. H. Mah, A. C. Tamhane, Detection of gross errors in process data, AIChE Journal 28 (5) (1982) 828–830.

[46] D. Ozyurt, R. Pike, Theory and practice of simultaneous data reconciliation and gross error detection for chemical processes, Computers and Chemical Engineering 28 (2004) 381–402.

[47] S. Qin, W. Li., Detection and identification of faulty sensors in dynamic process, AIChE Journal 47 (7) (2001) 1581–1593.

[48] F. Qi, B. Huang, Estimation of distribution function for control valve stiction estimation, Journal of Process Control 28 (8) (2011) 1208–1216.

[49] A. Dempster, A Generalization of Bayesian Inference, Jounral of the Royal Statistical Society. Series B 30 (2) (1968) 205–247.

[50] G. Shafer, A Mathematical Theory of Evidence, Princeston University Press, 1976.

[51] R. Gonzalez, B. Huang, Control loop diagnosis with ambiguous historical operating modes: Part 1. A proportional parametrization approach, Journal of Process Control 23 (4) (2013) 585–597.

[52] R. Gonzalez, B. Huang, Control loop diagnosis from historical data containing ambiguous operating modes: Part 2. Information synthesis based on proportional parameterization, Journal of Process Control 23 (4) (2013) 1441–1454.

[53] R. Gonzalez, B. Huang, Data-driven diagnosis with ambiguous hypotheses in historical data: A generalized Dempter-Shafer approach, in: 16th International Conference on Information Fusion, 2013.

[54] R. Gonzalez, B. Huang, Control-loop diagnosis using continuous evidence through kernel density estimation, Journal of Process Control 24 (5) (2014) 640–651.

[55] J. Venn, The Logic of Chance, MacMillan And Company, 1866.

[56] K. B. Korb, A. E. Nicholson, Bayesian Artificial Intelligence, Chapman & Hall/CRC, first edn., 2004.

[57] P. de Laplace, A Philosophical Essay on Probabilities, Dover, 1820.

[58] D. Fink, A Compendium of Conjugate Priors, Tech. Rep., Montana State University, Department of Biology, Bozeman Montana, 59717, 1997.

[59] S. Wilks, Mathematical Statistics, Wiley, 1962.

[60] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B (Methodological) 39 (1) (1977) 1–38.

[61] E. Parsen, On Estimation of a Probability Density Function and Mode, The Annals of Mathematical Statistics 33 (3) (1962) 1065–1076.

[62] D. Scott, Multivariate Density Estimation: Theory, Practice, and Visualization, Wiley, New York, first edn., 1992.

[63] G. Terrel, D. Scott, Variable Kernel Density Estimation, The Annals of Statistics 20 (3) (1992) 1236–1265.

[64] B. Efron, Bootstrap methods: Another look at the jackknife, Annals of Statistics 7 (1979) 1–26.

[65] J. Downs, E. Vogel, A plant-wide industrial process control problem, Computers and Chemical Engineering 17 (1993) 245–255.

[66] W. Ku, R. Storer, C. Georgakis, Georgakis. Disturbance detection and isolation by dynamic principal component analysis, Chemometrics and Intelligent Laboratory Systems 30 (1995) 179–196.

[67] T. McAvoy, N. Ye, Base control for the Tennessee Eastman problem, Computers and Chemical Engineering 18 (1994) 383–413.

[68] L. Chiang, E. Russell, R. Braatz, Fault diagnosis in chemical processes using fisher discriminant analysis, discriminant partial least squares, and principal component analysis, Chemometrics and Intelligent Laboratory 50 (2000) 243–252.

[69] N. Ricker, Decentralized control of the Tennesse Eastman challenge process, Journal of Process Control 6 (1996) 205–221.

[70] Tennessee Eastman challenge archive, URL `http://depts.washington.edu/control/ LARRY/ TE/ download.html`, 2011.

[71] S. Ahmed, B. Huang, S. Shah, Validation of continuous-time models with delay, Chemical Engineering Science 64 (3) (2009) 443–454.

[72] R. Fagin, J. Halpern, A new approach to updating beleifs, in: Uncertainty in Artificial Intelligence, 6 North-Holland, Amsterdam, 347–374, 1991.

[73] M. Wand, M. Jones, Kernel Smoothing, Chapman & Hall/CRC, first edn., 1995.

[74] I. Abramson, On Bandwidth Variation in Kernel Estimates - A Square-Root Law, The Annals of Statistics 10 (4) (1982) 513–525.

[75] A. Hyvarinen, E. Oja, Independent Component Analysis: Algorithms and Applications, Neural Networks 13 (4–5) (2000) 411–430.

[76] Fast ICA package for MATLAB and Octave, URL `http://research.ics.aalto.fi/ica/ fastica/`, 2000.

[77] E. A. Nadaraya, On Estimating Regression, Theory of Probability and Its Applications 9 (1) (1964) 141–142.

[78] T. Duong, M. L. Hazelton, Plug-in bandwidth matrices for bivariate kernel density estimation, Journal of Nonparametric Statistics 15 (2003) 17–30.

[79] T. Duong, M. L. Hazelton, Cross-validation Bandwidth Matrices for Multivariate Kernel Density Estimation, Scandinavian Journal of Statistics 32 (2005) 485–506.

Appendix A

Code for Kernel Density Regression

While kernel density regression has been presented Chapter 10 in a relatively simple manner, there are a number of safeguards and efficiency schemes that need to be put in place. As such, the code is too complex to be considered in standard chapter material and is instead, presented in the appendix.

A.1 Kernel Density Regression

This section contains the overall code for kernel density regression (it works for both standard and adaptive kernels) with both zeroth order (`fKernelRegFirst(x,Y,KDEx)`) and first order (`fKernelRegZeroth(x,Y,KDEx)`) being considered. In order to increase speed, a three-dimensional matrix toolbox was constructed and various functions from that toolbox were used (the functions are explained in Section A.2). Such functions are relatively easy to identify as they start with the character `z`.

Zeroth-order kernel density regression

```
function y = fKernelRegZeroth(x,Y,KDEx)

X  = KDEx.data;
Zt = KDEx.bwmZt;

%Data has nD rows, each are horizonal entries of x
[nX,p] = size(x);
[nD,¬] = size(X);
[¬,¬,nz] = size(Zt);
y = zeros(nX,length(Y(1,:)));

%Calculate kernel density (can take both static and variable kernels)
if nz == 1 %The usual case, for non—adaptive kernels
    clear('Weight');
    for n = 1:nX
        Xd = (X — ones(nD,1)*x(n,:))';
        Z = Zt*Xd;
        Weight(1,:) = exp( — 0.5*(sum(Z.^2,1)) );
        y(n,:) = (Weight*Y)/sum(Weight);
    end
else %If the kernel is adaptive, we have to multiply by different matrices
    clear('Weight')
    for n = 1:nX
```

```
        %entries are column vectors strung out depth-wise
        Xd = zcols((X - ones(nD,1)*x(n,:))');
        Z = z_matmultiply(Zt,Xd);
        Weight(1,:) = exp(- 0.5*sum(Z.^2,1));
        y(n,:) = (Weight*Y)/sum(Weight);
    end
end
```

First-order kernel density regression

```
function y = fKernelRegFirst(x,Y,KDEx)

X  = KDEx.data;
Zt = KDEx.bwmZt;

%Data has nD rows, each are horizonal entries of x
[nX,p] =  size(x);
[nD,py] = size(Y);
[~,~,nz] = size(Zt);
y = zeros(nX,py);

%Calculate kernel density (can take both static and variable kernels)
if nz == 1 %The usual case, for non-adaptive kernels
    clear('Weight');
    for n = 1:nX
        %Obtain Weights
        Xd = (X - ones(nD,1)*x(n,:))';
        Z = Zt*Xd;
        Weight(1,1,:) = exp( - 0.5*(sum(Z.^2,1)) );

        %Obtain regression denominator
        Z = [ones(1,1,nD);zcols(Xd)];
        oZ = z_matmultiply(Z,z_transpose(Z));
        Den = sum( z_matmultiply(Weight,oZ), 3);

        %Obtain regression numerator
        oY = z_matmultiply(Z,zrows(Y));
        Num = sum( z_matmultiply(Weight,oY), 3);

        B = Den\Num;

        y(n,:) = B(1,:);
    end
else %If the kernel is adaptive, we have to multiply by different matrices
    clear('Weight')
    for n = 1:nX
        %entries are column vectors strung out depth-wise
        Xd = zcols((X - ones(nD,1)*x(n,:))');
        Z = z_matmultiply(Zt,Xd);
        Weight(1,1,:) = exp(- 0.5*sum(Z.^2,1));

        %Obtain regression denominator
        Z = [ones(1,1,nD);Xd];
        oZ = z_matmultiply(Z,z_transpose(Z));
        Den = sum( z_matmultiply(Weight,oZ), 3);

        %Obtain regression numerator
        oY = z_matmultiply(Z,zrows(Y));
        Num = sum( z_matmultiply(Weight,oY), 3);

        B = Den\Num;
```

```
52
53          y(n,:) = B(1,:);
54      end
55  end
```

A.2 Three-dimensional matrix toolbox

MATLAB matrix operations only support two-dimensional matrices, but higher-dimensional equivalents can be implemented through element-wise multiplication and summation which support higher dimensions.

Multiplying Matrices

In this work, matrix manipulation must be done repeatedly for each element of data used in the kernel density estimate. Consider Figure A.1, on one side, there is a typical 2D matrix multiplication, but on the other size, there are two 3D matrices, where 2D matrix multiplication is to be repeated over the z axis.

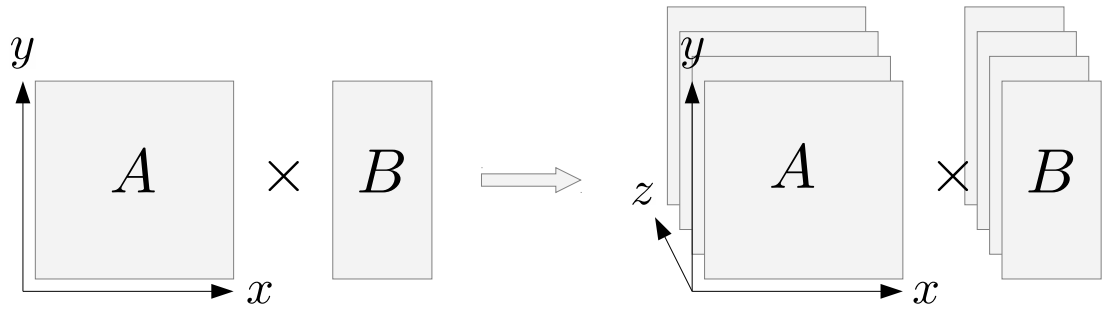


Figure A.1: `z_matmultiply`

The simple way to do this is to use a for loop

```
1  for z = 1:length(A(1,1,:))
2      C(:,:,z) = A(:,:,z)*B(:,:,z);
3  end
```

However, this method is slow. What is desired is to replace the for loop with this statement that can be executed efficiently

```
1  C = z_matmultiply(A,B)
```

This is done by the following function:

```
1  function C = z_matmultiply(A,B)
2
3  [ma,na,oa] = size(A);
4  [mb,nb,ob] = size(B);
5
6  %minimize for loops by looping the smallest matrix dimension
7  C = zeros(ma,nb,oa);
8
9  %if na ≠ mb || oa ≠ ob
10 %    fprintf('\n z_matmultiply warning: Matrix Dimmensions Inconsistent \n')
```

```
%else


if na == 1 && mb == 1 %outer product of two matrices
    C = repmat(A,[1,nb]).*repmat(B,[ma,1]);
elseif ma == 1 && na == 1; %A is a set of scalars
    C = repmat(A,[mb,nb]).*B;
elseif mb == 1 && nb == 1; %B is a set of scalars
    C = A.*repmat(B,[ma,na]);
elseif ma > nb %rows of A less than columns of B
    for j = 1:nb
        Bp = zeros(nb,mb,ob);
        Bp(j,:,:) = B(:,j,:);
        C(:,j,:) = sum(A.*repmat(Bp(j,:,:),[ma,1]),2);
    end
else          %rows of B less than columns of A
    for i = 1:ma
        Ap = zeros(na,ma,oa);
        Ap(:,i,:) = A(i,:,:);
        C(i,:,:) = sum(repmat(Ap(:,i,:),[1,nb]).*B,1);
    end
end

end
```

Rearranging Matrices

Some code has also been used for rearranging 3D matrices. For example, for taking the transpose of all depth-wise matrices (as seen in Figure A.2) is obtained using the function `z_transpose`

```
function Xt = z_transpose(X)
[nx,ny,nz] = size(X);

Xt = zeros(ny,nx,nz);

if nx < ny
    for xi = 1:nx
        Xt(:,xi,:) = X(xi,:,:);
    end
else
    for yi = 1:ny
        Xt(yi,:,:) = X(:,yi,:);
    end
end
```

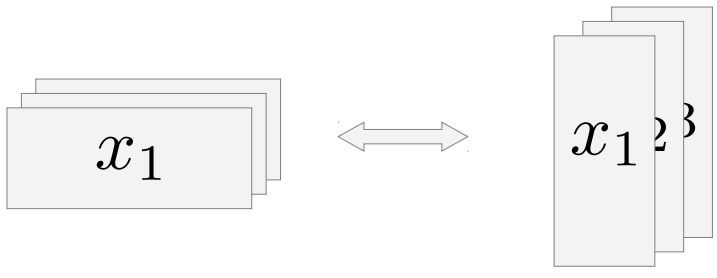
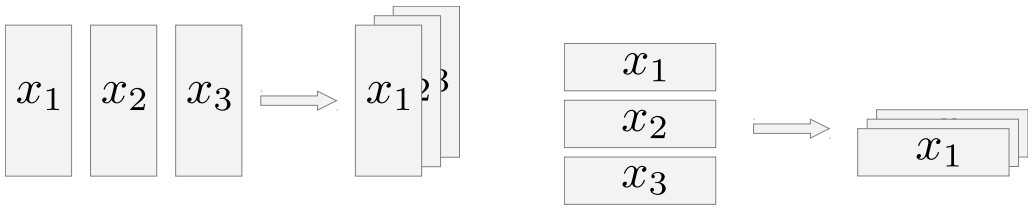


Figure A.2: `z_transpose`

In addition, matrices can be converted into n depth-wise columns or n depth-wise rows using `zcols` and `zrows`.

```
1  function Azr = zrows(A)
2  Azr(1,:,:) = A';
3  end
4
5  function Azc = zcols(A)
6  Azc(:,1,:) = A;
7  end
```



(a) `zcols` (b) `zrows`

Figure A.3: Converting matrices depth-wise